NR INDEKSU 353965
PL ISSN 0860-1674

Z MIKROKOMPUTEREM NA TY

10

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 10(46) PAŹDZIERNIK 1989 CENA 500 ZŁ

ULTIMATUM DLA „GIEŁDY BAJTKA"

WIELKA UNIFIKACJA

SIDEKICK

SZLACHETNE ZDROWIE

STANISŁAW OSTOJA-KOTKOWSKI

SZTUKA KOMPUTEROWA S.32



TOTAL ECLIPSE

KOMPUTEROWY AIDS?

Wbrew pesymistycznym przewidywaniom nie doszło do katastrofy. Mam na myśli katastrofę, która — jak zapowiadano — miała nastąpić 13 października, a jej ofiarą miały stać się zespoły pamięci milionów IBM-podobnych komputerów na całym świecie. Oblicza się, że komputerów tej klasy jest ponad 23 miliony sztuk. Właśnie 13 października miał się uaktywnić wprowadzony wcześniej do obiegu wirus, którego zadaniem — tak w każdym razie kazała wierzyć rozpowszechniona wśród programistów całego świata plotka — miało być spowodowanie maksymalnego spustoszenia w centralnych zespołach pamięci.

13 października przeszedł jednak w miarę spokojnie. Jak informowała następnego dnia Polska Agencja Prasowa, zapowiadanej „infekcji" uległ tylko niewielki procent komputerów. W Wielkiej Brytanii, na przykład, wirus wyrządził poważne szkody w pamięci komputera Królewskiego Instytutu Niewidomych. Najwięcej mikrokomputerów zostało zaatakowanych w Holandii (ok. 50), kilka w Portugalii i kilka (zaledwie kilka!) w USA.

Wstępne meldunki po dniu „X" świadczyły więc o tym, że większość użytkowników komputerów wyszła z zapowiadanej apokalipsy obronną ręką. Świadczyć to może o tym, że plotka o „złośliwości" i „przenikliwości" nowej grupy wirusów — datacrime — była przesadzona. Ale nie sposób też nie docenić wpływu świadomych przeciwdziałań. 13 października stał się bowiem jednocześnie dniem triumfu twórców i producentów programów przeciwwirusowych. Nie jest zresztą wykluczone, że to właśnie oni zasiali panikę wśród posiadaczy IBM-ów!

Cała ta historia związana z zapowiadanym na 13 października pogromem informatycznym, skłania do poczynienia kilku refleksji.

Po pierwsze — okazuje się (w każdym razie nie ma powodów aby sądzić, że jest inaczej), iż w świecie komputerów pojawiły się wprawdzie różne „wirusy" i „bakterie", ale są to na szczęście szczepy łagodne, na które dość łatwo jest wyprodukować skuteczną „surowicę". Nie pojawił się jeszcze komputerowy odpowiednik wirusa HIV, dzięki czemu nie grozi nam jeszcze informatyczny AIDS! Należy się tylko z takiej sytuacji cieszyć, gdyż tak wiele już w naszym życiu zależy od komputerów, że paraliż głównych systemów informatycznych decydujących o dzisiejszym kształcie życia na naszej planecie byłby zarazem paraliżem cywilizacji.

Po drugie — okazało się przy okazji 13 października, jak bardzo jesteśmy podatni (w naszym naukowo-technicznym wieku!) na wszelkie plotki, jak wąska granica dzieli to, co możliwe i racjonalne, od zachowań irracjonalnych. Bo przecież za irracjonalne należy jednak uznać przekonanie, że pirackie wirusy mogą okazać się silniejsze od wysiłku milionów programistów, włożonego w przygotowanie i zabezpieczenie olbrzymich zbiorów programów użytkowych.

Ale może ta odrobina irracjonalizmu jest nam niezbędna? Może to jest właśnie główną przyczyną, dla której nad kolejnymi miastami naszego globu (ostatnio tego zaszczytu dostąpił leżący w centralnej Rosji Woroneż) kolejni świadkowie wykrywają błąkające się po niebie i lądujące w miejskich parkach UFO?

Waldemar Siwiński

ZA MIESIĄC

WIELKI KONKURS ŚWIĄTECZNY! GŁÓWNA NAGRODA KOMPUTER!

- Nowe opisy gier Westroyer, BMX Simulator +
- Tetris na Spectrum!
- Schemat generatora trójkanałowego do Spectrum z opisem
- Najlepszy DOS dla Atari
- Dostałem Amigę — podstawowe informacje dla posiadaczy tego komputera
- Intel rzuca wyzwanie IBM-owi
- W Joysticklandzie
- Komiks z komputera

Bajtek

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY:
Waldemar Siwiński — kierownik zespołu „Bajtka"
Marek Czarkowski — sekretarz redakcji „Bajtka"
Roman Poznański — kierownik działu klanów
Wanda Roszkowska — opr. graficzne

KLANY REDAGUJĄ:
Atari — Wojciech Zientara
Amstrad — Jonasz Mayer
Commodore — Klaudiusz Dybowski, Dominik Falkowski
Spectrum — Marian Przasnyski

STALE WSPÓŁPRACUJĄ:
Andrzej Pilaszek, Janusz Jarmoch, Marcin Borkowski, Łukasz Czekajewski, Waldemar Nowak, Michał Sobieszuk, Maciej Pietras, Marcin Bójko, Jacek Kunowski, Michał Karkuciński

Zdjęcia w numerze Leopold Dzikowski

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wółtańska

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.

[illegible] zł.

Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.

Nr zlecenia 12439 n. 150 000 egz.
A-111

SM SZTANDAR MŁODYCH

ULTIMATUM DLA GIEŁDY „BAJTKA"



Z informacji, że w każdą sobotę i niedzielę odbywa się w Warszawie giełda komputerowa zwana Giełdą „Bajtka", wynika niewiele. Wystarczy jednak stawić się u wejścia do szkoły nr 25 na rogu Grzybowskiej i Marchlewskiego w sobotnie południe i zobaczyć to na własne oczy.

12.30–13.30 OCZEKIWANIE

Samochodami prywatnymi i taksówkami zajeżdżają stali bywalcy, posiadający rezerwację stanowisk. Przywożą góry sprzętu i ogromne pliki pieniędzy. Ustawiają się w grupki, rozmawiają. Co zapobiegliwsi rozkładają swój towar na ławce przed szkołą i na gazetach, na asfalcie. Z reguły, nie wchodząc na teren giełdy zakupić można telefon bezprzewodowy, każdą ilość dyskietek, kolorową koszulkę, cartridge do MSX.

Powoli, bez pośpiechu wlewają się handlarze na teren szkoły. Okazują przy wejściu rezerwacje i targają kilogramy sprzętu po schodach, spiesząc do swych stoisk.

Tam rozwijają kramiki. Ustawiają sprzęt na ławkach wieszają wymyślne szyldy, płachty poklejonych reklamówek gier, wyciętych z zachodnich czasopism. Majsterkowicze montują przemyślne rusztowania z listewek i imadeł lub stojaka do suszarki. Na rusztowaniach tych wieszają efekty całotygodniowego kopiowania, rysowane przez całą noc plansze z tytułami i wydrukami nowości.

Nagrywane w pocie czoła stosy kaset z zestawami gier, opatrzone okładką „Computer Studio" rozkładają na zwykłych, szkolnych ławkach. Kasety przylepiają scotchem. Obok rozkładają katalogi, wystukiwane na maszynie lub drukowane na drukarce. Niektóre, cenniejsze katalogi przywiązywane są sznurkiem do nogi stołowej.

Tymczasem pod szkołą zbierają się pomniejsi handlarze, którzy nie dorobili się jeszcze rezerwacji. I szybko się nie dorobią, gdyż organizator giełdy już od dawna nie dysponu-





GRA O JUTRO

je wolnymi stoiskami. Miłościwi posiadacze rezerwacji mogą jednak odstąpić ją komuś, za sumę ok. pół miliona (15.08.89, proszę przeliczyć).

13.30 SZARŻA

Otwierane są drzwi, na które od pół godziny napiera falujący, zbity tłum amatorów wolnych miejsc na III piętrze, które nie jest objęte rezerwacją. Również sala gimnastyczna, gdy otwarta, jest obiektem ataku hordy handlarzy.

Zanim jednak drzwi staną otworem, nierzadko pęka szyba, łamie się komputer (szczególnie Amstrad 6128), wybucha monitor, trzeszczą żebra.

Wreszcie szaleńczy bieg na górę. Po drodze nie wolno upaść, gdyż zostanie się stratowanym. Komuś kiedyś rozsypały się w biegu dyskietki. Po chwili nie miał co zbierać. Niebezpieczny jest zakręt na I piętrze, w drodze na salę gimnastyczną. Na śliskiej posadzce można się łatwo przejechać i z hukiem zatrzymać na ścianie z tablicą „Preorientacja zawodowa".

Brakuje miejsca. Dlatego sprzęt rozkładany jest wszędzie. Na parapetach z kartką „Zakaz rozkładania sprzętu w tym miejscu", na podłodze i przyniesionych stolikach. Kto tylko sprzedaje, ustawia się w szpaler biegnący środkiem korytarza na I i II piętrze. Spóźnieni chodzą tam i z powrotem z kartkami na plecach, brzuchu lub transparentami „Kupię bony", „Sprzedam C128" itp.

13.35–17.00 MARKET SQUARE

Tuż za szarżującymi handlarzami podążają zainteresowani. Kto to? Zwykli ludzie, przyzwyczajeni płacić za wszystko (pierwszą opłatę uiścili przy wejściu). Dzieci z rodzicami w poszukiwaniu nowych gier. Młodzież pragnąca uzupełnić zapas instrukcji, mapek, programów. Poważni dorośli w pogoni za częściami elektronicznymi, przystawkami do komputera. Całe rodziny, chcące zakupić komputer. Oprócz tego dzieciaki, cinkciarze, kieszonkowcy i ci, którzy przychodzą na giełdę, by nie wypaść z obiegu.

Złotówka już dawno przestała być walutą kupna-sprzedaży w transakcjach powyżej, przyjmijmy za wskaźnik, 100.000 zł (10 dolarów na dziś). Komputery, monitory, stacje dysków, drukarki, za to wszystko ponurzy sprzedający wołają 160, 200, 500 nie podając nawet sakramentalnego słówka „papierów". Handel samymi dolarami też kwitnie, coraz więcej ludzi kręci nosami na bony.

Za złotówki można kupić kasety, programy, joysticki, dyskietki, gumy do żucia itp. Co jeszcze? Zdarzają się buty, koszulki „T-shirts", czasopisma włącznie z Playboyem, książki szkolne i odbijane na ksero instrukcje i opisy programów, papier do drukarki, kable, kasety video a także „sennik egipski" i „Tajemnice uzdrowicieli". Sprzęt hi-fi, video, compact dyski oczywiście za dolary.

Są więc handlarze więksi i mniejsi. Więksi po roku działalności na giełdzie urywają się na stałe na Zachód, by stamtąd wspomagać sprzętem kolegów. Mniejsi awansują, zarabiają, ale wciąż tkwią w swym kręgu.

Nie przeczę, że wśród handlarzy można znaleźć bardzo zdolnych hackerów, programistów z ogromną wiedzą. Ich umiejętności przeliczane są na pieniądze jedne lub drugie, wyciągane z kieszeni klientów.

Gros klientów handlarzy mniejszego kalibru stanowią dzieci (przepraszam, do 15 roku życia), które za nową grę dadzą sobie uciąć rękę. Na razie jednak zapuszczają ją do kieszeni rodziców, by mieć czym zadowolić handlarza, który albo całą noc nagrywał zestawy, albo zrobi to na miejscu. Nikogo nie interesuje, skąd są nowe gry. Ma je jeden, po tygodniu wszyscy. Ale gra nie wyjdzie poza krąg handlarzy. Oni wyznają zasadę „jeden za jeden", lub częściej „tysiąc za jeden".

Co ciekawsze, gdy zdolniejszy hacker giełdowy napisze grę program muzyczny lub użytkowy, to za jego kopiowanie jest w stanie pobić, nasłać kolesiów lub przewrócić stolik ze sprzętem. Program swój sprzedaje kolegom-handlarzom po przystępnej cenie z pięcioma zerami i zastrzeżeniem, by go nie „puszczali" dalej. I tak wszyscy są zadowoleni.

Często spotyka się obrazek: do stoiska podchodzi dzieciak z „Bajtkiem". Odczytuje z namaszczeniem tytuły gier, których nie ma, zaś handlarz notuje je skrzętnie. Potem nagrywa je dzieciakowi na taśmę, inkasuje pieniądze, liczy kasę i proceder toczy się dalej.

Ciekawość budzą dwa PC: Amstrad 1512 i jakiś dalekowschodni klon. Urzędują przy nich nieopierzeni uczniowie SP i nagrywają programy poważnym, dorosłym ludziom. Idzie wszystko: edytory, nowe wersje DOS-u, „toolsy", programy graficzne, bazy danych, coraz liczniejsze gry. Cena jest stała: 3 000 zł za nagranie dyskietki. O tym, że na trzech dyskietkach klient otrzymuje to, co zwykle mieści się na dwóch, wie mało kto. Najmniej płacący, gdyż z reguły nie zna się on na PC i szuka kontaktów i perspektyw na giełdzie.

17.00–19.00 ZWIJAMY INTERES

Kto miał jakieś sprawy, już je załatwił. Większość transakcji doszła do skutku. Handlarze liczą utarg, wymieniają uwagi. Umawiają się na spotkania w środku tygodnia. Nikt z zewnątrz nie zajrzy. Bo i po co? O 18.30 w środku nie ma już nikogo. Pozostają papierki po cukierkach, czekoladach, puste butelki po Coli (400 zł w bufecie), śmieci, rozrzucone ławki i krzesła.

REFLEKSJE

Czy złożony proceder, jaki uprawiany jest na ul. Grzybowskiej nie pozostaje w kolizji z prawem? Nie, ponieważ nie istnieje w Polsce prawo dotyczące ochrony praw autorskich programów. Cała reszta jest prawie legalna, nie mówiąc o tym, że cała giełda tchnie dziadostwem, prowizorką i pchlim targiem. Czy tak wyglądać ma rynek komputerowy w Polsce? Technika śliny i plastra dominuje na razie w tym zlepku cwaniactwa i nie kwestionowanych umiejętności. Spieszymy z wyjaśnieniem, że nazwa „Giełda Bajtka" jest pozostałością z czasów gdy nasza redakcja wraz z innymi organizacjami, którym zależało i nadal zależy na podnoszeniu kultury informatycznej w Polsce, zgłosiła projekt utworzenia giełdy. Informujemy, że z faktu patronatu nad Giełdą nie mamy żadnych korzyści materialnych. Nie identyfikujemy się już z zachodzącymi tam zajwiskami. Nie musimy chyba dodawać, że za uprawianie wyżej wymienionych procederów handlarze, gdyby byli np. w Anglii, długo nie wyjrzeliby z wilgotnych lochów Tower. Ale u nas egzystują w dobrobycie i co gorsza, poważaniu.

(red.) Marcin Przasnyski

W związku z powyższym Redakcja „Bajtka" stawia organizatorom Giełdy następujące warunki.

1. Należy zlikwidować handel dolarami, markami, funtami i innymi walorami dewizowymi na Giełdzie (W Polsce handel dewizami nie jest już karalny, karalne natomiast jest nie płacenie podatków od zysków z tytułu prowadzenia działalności finansowej).

2. Ograniczyć do minimum tzw. rezerwację stoisk, nie mającą nic wspólnego z daniem równych szans wszystkim chcącym sprzedawać swoje programy czy sprzęt.

3. Należy uruchomić kącik bezpłatnych porad dla kupujących, w którym każdy mógłby dowiedzieć się np. czy warto kupić oferowany przez sprzedawcę komputer, albo ile warto zapłacić za program.

4. Bezwzględnie należy zakazać sprzedaży odbitek kserograficznych zachodnich publikacji poświęconych informatyce. Jest to naruszenie praw autorskich w świetle prawa obowiązującego w naszym kraju a więc przestępstwo.

5. Wiemy, że administracyjnymi zakazami nie zlikwiduje się zjawiska piractwa w dziedzinie oprogramowania tak rozpowszechnionego w Polsce. Dlatego zmuszeni jesteśmy pogodzić się z faktem jego istnienia. Proponujemy jednak wprowadzenie zniżkowych opłat dla sprzedających własne (napisane przez siebie) oprogramowanie.

W przypadku nie spełnienia powyższego ulimatum w terminie dwóch miesięcy od ukazania się tego numeru, Redakcja Bajtka wycofa się z patronatu nad Giełdą i odbierze organizatorom prawo używania nazwy „Giełda Bajtka". O sprawie poinformujemy Czytelników.

Znudzeni grami komputerowymi, długim czasem oczekiwania na wczytanie programu z magnetofonu i niewyraźnym obrazem na ekranie telewizora, po pewnym czasie zaczynamy zastanawiać się, jak inaczej wykorzystać nasz komputer i jak polepszyć komfort korzystania z niego. Wśród wielu potencjalnych zakupów rozważamy często nabycie stacji dysków. Właśnie z nią wiążą się największe nadzieje na poprawę naszej sytuacji.

POTRZEBA STANDARDU

W dziedzinie 16-bitowych komputerów ogólnie przyjętym w Polsce standardem jest IBM PC pracujący pod kontrolą dyskowego systemu operacyjnego MS—DOS. W przypadku sprzętu 8-bitowego sytuacja jest bardziej skomplikowana. Praktycznie każdy producent oferuje inny system operacyjny na swoje komputery. Stan taki utrudnia rozwój oprogramowania, ponieważ pojawienie się na rynku nowej maszyny wymaga tworzenia wszystkich przewidzianych na nią programów od nowa. Oczywiście podnosi to koszty całego przedsięwzięcia i utrudnia współpracę nowych systemów ze starymi. Problemem jest na przykład przeniesienie danych i programów z poprzednio używanego komputera. Kupując nowy sprzęt praktycznie musimy uczyć się wszystkiego od początku.

W świecie 8-bitowych komputerów, wyposażonych w procesor Zilog Z80 lub Intel 8080, pewnym rozwiązaniem przedstawionych problemów jest system operacyjny CP/M. Z punktu widzenia użytkownika różni się on niewiele od popularnego na IBM PC MS—DOSa, którego jest bezpośrednim prekursorem. Wiele standardów oprogramowania CP/M'u, takich jak Turbo Pascal, dBase, Wordstar, Multiplan i GSX, zostało adaptowanych na IBM'a.

CP/M może być w Polsce dla domowych komputerów 8-bitowych tym samym, czym jest MS—DOS dla profesjonalnego sprzętu 16-bitowego.

KOMPUTERY CP/M'u.

Wbrew pozorom z CP/M'u można korzystać na wielu komputerach dostępnych na naszym rynku.

— **Amstrad PCW 8256/8512** to nie tylko specjalizowany procesor tekstów, to także najlepszy, seryjnie produkowany komputer CP/M'u Plus. W swojej klasie wyróżnia go duża pamięć operacyjna 256—512 KB, grafika wysokiej rozdzielczości — 720×256 punktów, i oferowana w systemie drukarka. Dyskietki 3" o pojemności 2×180KB. Uzupełnienia: napęd 5 i 1/4", dysk twardy 20MB, Centronics, RS 232C, zegar czasu rzeczywistego.

— **Amstrad CPC 6128** ma świetny BASIC dostępny w systemie AMSDOS, ale jednocześnie pozwala na korzystanie zarówno z CP/M Plus, jak i CP/M 2.2. Brak ramdysku i 7-bitowy interface równoległy to wady tego komputera. Pamięć RAM 128KB, dyskietki 3" 2×180KB, rozdzielczość 640×200. Uzupełnienia: napęd 5 i 1/4", dysk twardy, 8-bitowy Centronics, RS 232C, ramdysk 64—256KB, catridge'e z oprogramowaniem w ROM'ie.

— **Amstrad CPC 464** — poprzednik modelu 6128, wyposażony w magnetofon kasetowy zamiast stacji dysków. Dodanie interface'u i napędu dyskowego umożliwia prace w systemie CP/M 2.2. Pamięć RAM 64KB, pozostałe parametry jak dla CPC 6128.

— **SpectraVideo SVI-738** — dalekowschodni komputer standardu MSX, sprzedawany przez Centralną Składnicę Harcerską, zakupiony głównie przez szkoły. Oprócz systemu MSX—DOS, oferowany jest CP/M 2.2. Firma pana Pawła Ciężkiego z Łodzi, zasłużona w tworzeniu i adaptacji oprogramowania na ten komputer, dostarcza uproszczoną wersję CP/M'u Plus. Pamięć RAM 64KB, RS 232C, Centronics, stacja dysków 3.5" 360 KB, rozdzielczość 192×256. Sprzedawany bez monitora. Uzupełnienia: stacja 5 i 1/4". Możliwe powiększenie rozdzielczości do 192×512.

— **Commodore 128D** — komputer kompatybilny z C-64. Wyposażony w drugi procesor Z80 pozwala na prace w CP/M Plus. Parametry podobne do Amstrada 6128: pamięć RAM 128 KB, rozdzielczość 640×200. Dyskietki 5 i 1/4 cala, Centronics, RS 232C. Sprzedawany bez monitora.

— **Junior** — niezrealizowana nadzieja naszego szkolnictwa. Oprócz zgodności z ZX Spectrum pozwala na pracę w sieciowym systemie CP/J, który z

WIELKA UNIFIKACJA
DLACZEGO CP/M?

punktu widzenia pojedynczego użytkownika odpowiada CP/M'owi 2.2. Pamięć RAM 64 KB, rozdzielczość 192×256 lub 192×512, napęd 5 i 1/4".

— **ZX Spectrum** — pierwszy najbardziej popularny komputer domowy w Polsce. Po dokupieniu portugalskiej stacji Timex'a, z jednym lub dwoma napędami umożliwia pracę w systemie CP/M 2.2. Pamięć RAM 64KB, rozdzielczość 192×512, napęd 3".

W śladowych ilościach występują CP/M'owe rozszerzenia do **Commodore'a C-64**, natomiast nie słyszałem o CP/M'ie na 8-bitowe Atari.

Rzadko widuje się w naszym kraju, popularne na Zachodzie przed kilku laty, profesjonalne komputery CP/M'u firm Osborne, Kaypro czy InterTec (SuperBrain).

OPROGRAMOWANIE CP/M'u

Komputer bez oprogramowania jest kupą bezużytecznego złomu. Pisanie programów jest inwestycją bardziej kosztowną niż produkcja nowych generacji sprzętu. O dużej sile przebicia CP/M'u decydują bogate i różnorodne oprogramowanie.

W dziedzinie edytorów tekstowych wiodącym standardem jest tutaj Wordstar. Zakładanie i obsługę baz danych umożliwia produkt firmy Ashton Tate — dBase II. Do sporządzania zestawień tabelarycznych służą programy Supercalc i Multiplan. Dane opracowane przy ich pomocy mogą być prezentowane w formie graficznej na ekranie komputera, na drukarce lub na ploterze przy pomocy programu DR. GRAPH. Do prac typu CAD (projektowanie wspomagane komputerowo) można stosować systemy przypominające AUTOCADa, a więc DR.DRAW, MicroDraft itp..

Problem grafiki rozwiązano w systemie CP/M wprowadzając jego graficzne rozszerzenie w postaci GSX'a (ang. Graphics System eXtension). Jest to pakiet procedur obsługujących w jednolity sposób, z punktu widzenia zastosowań graficznych, wszystkie urządzenia wyjściowe, takie jak ekran komputera, drukarki i plotery. GSX jest w zasadzie narzędziem dla programisty, ale korzystają z niego także programy DR DRAW i DR GRAPH.

Bogato reprezentowane są w CP/M'ie różne języki programowania. Do najbardziej popularnych należy obecnie Turbo Pascal, wspierany dodatkowymi bibliotekami procedur. Komfortowe warunki pracy przy programowaniu w asemblerze zapewnia pakiet M80, LINK i DDTZ. Miłośnicy FORTRANu IV mogą korzystać z kompilatora F80, a osoby zainteresowane językiem BASIC mają do dyspozycji interpreter Mallard BASIC'a i kompilator firmy Microsoft. Kolejne ciekawe produkty to AZTEC C i Pascal MT+.

WYMIANA OPROGRAMOWANIA I DANYCH.

Komputery muszą rozmawiać ze sobą. Konieczne jest zapewnienie przenoszalności danych i programów między różnymi komputerami. Przykładowo, możemy wykorzystywać ZX Spectrum do rejestracji wyników eksperymentu, który będziemy opracowywać na IBM PC. Dane w terenie, zbierane przy pomocy przenośnego Z88, mogą być przeniesione na inny komputer. Artykuł napisany na Amstradzie redakcja wydawnictwa powinna mieć szansę przygotować do druku, posługując się IBM'em.

Wymiana zbiorów między komputerami może odbywać się na dwa sposoby. Jednym z nich jest wykorzystanie dyskietek jako nośnika informacji. Dominującym standardem w Polsce jest format 5 i 1/4". Z tego powodu każdy komputer CP/M'u powinien być wyposażony w taki napęd lub powinna istnieć możliwość jego dołączenia. Niestety, nawet w CP/M'ie, dyskietki o tym samym rozmiarze różnią się ilością ścieżek, sektorów na ścieżkę i innymi parametrami. Z tego powodu, razem z komputerem należy dostarczyć oprogramowanie pozwalające na posługiwanie się wieloma obcymi formatami. Ze względu na powszechność sprzętu typu IBM PC konieczne jest również uwzględnienie tego standardu.

Inną metodą wymiany informacji między komputerami jest wykorzystanie złącza szeregowego typu RS 232C i ewentualnie modemu. Kolejnym zaleceniem jest więc wyposażenie komputera w to złącze i dostarczenie odpowiedniego oprogramowania komunikacyjnego. W CP/M'ie, w prostych przypadkach transferu zbiorów tekstowych możemy posługiwać się nierezydentną komendą PIP. Niestety, w niektórych systemach (SpectraVideo), polecenie to nie pozwala na korzystanie z interface'u szeregowego. Alternatywne i profesjonalne podejście do problemu zapewnia użycie programu KERMIT, będącego kolejnym międzynarodowym standardem możliwym do implementacji na każdym komputerze wyposażonym w interface szeregowy.

Reasumując, możemy sformułować dwa minimalne rozwiązania sprzętowo—programowe umożliwiające wymianę informacji między komputerami:

1. Napęd 5 i 1/4" + oprogramowanie obsługi dyskietek IBM PC
2. Złącze RS 232C + program KERMIT

STANDARTY DO ADAPTACJI

Omawiany powyżej program KERMIT, dostępny przede wszystkim na komputer IBM PC i większe maszyny, jest jednym ze standardów godnych upowszechnienia na 8-bitowym sprzęcie CP/M'u. Pozostałe programy zasługujące na uwagę to: **XTREE, Norton Utilities, CHIWRITER i AUTOCAD.**

XTREE jest nakładką systemową pozwalającą na wygodną obsługę plików na dyskietkach. Umożliwia on oglądanie, kasowanie, kopiowanie zbiorów oraz wiele innych operacji.

Norton Utilities to program typu „Disk Doctor" pozwalający obejrzeć i zmienić zawartość dowolnego sektora dyskowego. Jedną z jego opcji jest odzysk skasowanych zbiorów.

CHIWRITER jest edytorem pracującym w trybie graficznym i zyskał dużą popularność w naszym kraju dzięki łatwości z jaką można wprowadzić w nim polskie litery, zarówno na ekranie, jak i na drukarce, bez żadnych przeróbek sprzętowych. Doceniany szczególnie przez ludzi piszących teksty naukowe, zawierające skomplikowane formuły matematyczne lub chemiczne.

AUTOCAD to znany na całym świecie produkt firmy AUTODESK. Używany przez profesjonalistów do projektowania wspomaganego komputerowo. Posiada chyba najbogatszą, w swej klasie, bibliotekę, gotowych procedur i elementów.

Każdy z tych programów może być zaadaptowany na 8-bitowych komputerach CP/M'u. W przypadku AUTOCADA, ze względu na ograniczenia sprzętowe, nie jest celowa pełna implementacja systemu, ale zapewnienie zgodności z formatem plików używanych na IBM PC. Udostępnienie użytkownikom CP/M'u wspomnianego oprogramowania podniesie istotnie komfort posługiwania się tymi komputerami i znacznie zwiększy ich możliwości.

PIENIĄDZE, CZYLI KTO ZA TO ZAPŁACI...

Bez zmrużenia oka płacimy 100, 200, 500 albo i więcej dolarów za potrzebny nam sprzęt komputerowy. Dlaczego mamy węża w kieszeni, kiedy chodzi o kupno programu?

Opisane w artykule oprogramowanie nie jest, w większości przypadków, powszechnie dostępne. Wręcz przeciwnie, należy je dopiero napisać lub adaptować. Prawdopodobnie nie zostanie to już zrobione przez zachodnich programistów, ponieważ tam łatwiej kupić sprzęt 16-bitowy niż odchodzące w przeszłość komputery 8-bitowe. U nas o zmianę samochodu czy komputera na nowy model jest znacznie trudniej.

Uważam, że w związku z tym powinniśmy „wycisnąć" z dostępnego sprzętu 8-bitowego maksimum tego, co się da. Kto jednak podejmie się stworzenia takiego oprogramowania, w warunkach naszego rynku, zdemoralizowanego doszczętnie przez piractwo?

Obawiam się, że nikt. Dopóki nie zmieni się mentalność ludzi kupujących za ciężkie pieniądze komputery, a lekceważących nakład pracy w stworzenie dobrego, profesjonalnego oprogramowania, dopóki nie zmienią się w tym kraju prawa dotyczące ochrony programów, dopóty będziemy mieli, tragiczny w skutkach, zastój w tej dziedzinie.

Uprawiana przez wszystkich w Polsce polityka Kalego: „Kali ukraść krowę — dobry uczynek, Kalemu ukraść — zły", musi ulec zmianie, jeśli myślimy poważnie o dalszym rozwoju komputeryzacji u nas.

PERSPEKTYWY

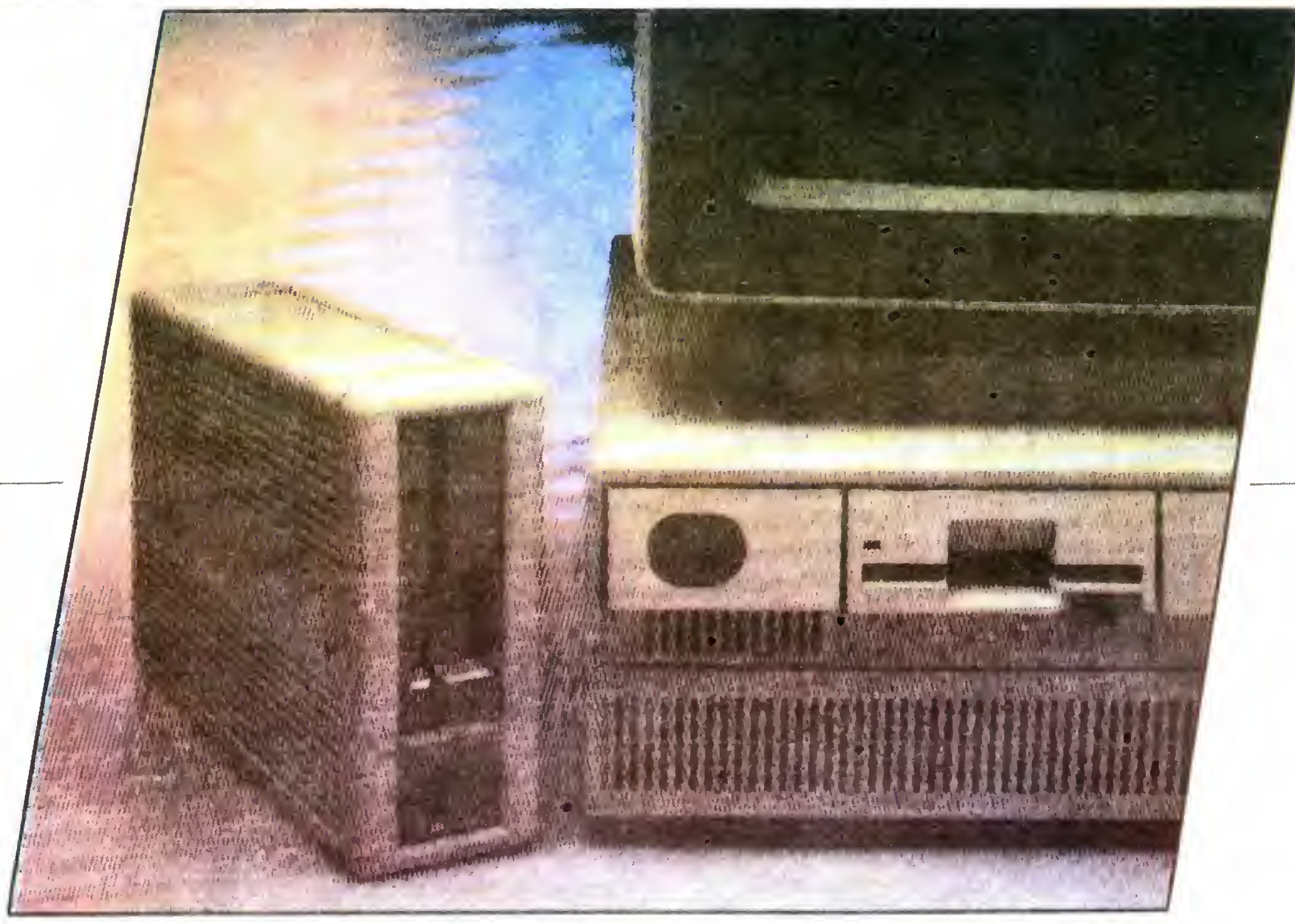
Jeśli przekonamy ludzi, że bardziej opłaca się kupować niż kraść programy, to perspektywy mogą być dobre.

Podobieństwo CP/M'u do MS DOS'a skłania do przypuszczenia, że osoby korzystające z niego nie będą miały większych trudności przy przejściu na bardziej nowoczesne komputery IBM PC. Osiągnięcie kompatybilności drugiego stopnia, tzn. zgodności idei i standardów, będzie także temu sprzyjać.

CP/M jest również wielką szansą polskiego szkolnictwa. W warunkach, w których nie możemy zapewnić naszym szkołom dużej ilości komputerów IBM PC współpracujących ze sobą w sieci typu Novell, musimy pamiętać, że większość sprzętu zakupionego przez oświatę pozwala na korzystanie z CP/M'u. Dotyczy to nawet ZX Spectrum, jeśli wyposażymy go w stacje dysków Timex'a.

Jonasz Mayer





3.5"–5.25"

Dla wszystkich użytkowników komputerów klasy PC pojawienie się nowej generacji PS/2 oznacza konieczność przeniesienia swoich zbiorów z dyskietek miłościwie nam panującego formatu 5.25". na przyjęte jako nowy standard dyskietki 3.5" W dniu w którym ostatnia stacja 5.25" powędruje do muzeum, problem automatycznie zniknie, ale zanim to nastąpi trzeba szukać rozwiązań na tak zwany okres przejściowy.

Tym którzy już kupili nowy komputer PS/2, i nerwowo zastanawiają się nad tym gdzie wetknąć posiadane dyskietki 5.25". przyszła z pomocą firma MicroWay produkując urządzenie o nazwie ADDON 36/12. Jest to przyłączalna do komputera zewnętrzna stacja dyskietek, zachowująca się (dla użytkownika) dokładnie tak samo jak stacje wbudowane w komputer ADDON 36/12 po podłączeniu do komputera sam (bez dodatkowej ingerencji użytkownika) instaluje się w systemie, oraz ładuje do pamięci operacyjnej komputera krótki program obsługi stacji. Urządzenie potrafi odczytywać dyskietki 360 KB i 1.2 MB, więc powinno zadowolić zarówno byłych posiadaczy XT jak i AT Konstruktorzy zadbali także o szybkość działania — dane przesyłane są z prędkością 500 Kb/s, znacznie szybciej niż w analogicznych urzadzeniach korzystających nie z bezpośredniego połączenia z pamięcią operacyjną, a z łącza szeregowego.

(mb)

PRZESTROGA DLA HACKERSÓW

W listopadzie 1988 roku, już w kilka godzin po tym, gdy skonstruowany przez niego wirus zaczął się rozprzestrzeniać, **Robert Morris** (na zdjęciu) ujrzał pierwsze efekty swego zwariowanego dowcipu: jego komputer w Uniwersytecie Cornell (USA) zablokował się. Sprawa ta stała się głośna na całym świecie, ale hacker-student poniósł jak najgorsze konsekwencje swego kawału. W momencie gdy oddawaliśmy ten numer do druku, sąd wstępny (decydujący w USA o przekazaniu sprawy do normalnego rozpoznania procesowego) uznał że 24-letni Robert jest winien nieuprawnionego włamania się do federalnej sieci komputerowej, powiązanej w dodatku z siecią ministerstwa obrony. Jeśli zostanie to udowodnione, Robert może zostać skazany na 5 lat więzienia i zadośćuczynienie wszystkich szkód wyrządzonych przez wirusa. A warto wiedzieć, że jedna z firm przemysłowych ocenia te szkody na 96 mln dolarów.

(ws)



KOMPUTER I DRINKI

Co powinien zrobić właściciel baru, aby jak najdłużej zatrzymać klientów, którzy wpadli tylko na małą godzinkę? Oprócz dobrych drinków i sprawnej obsługi, najlepiej zaproponować jakieś miłe, a równocześnie wciągające zajęcie. Sam pomysł nie jest nowy, przykładem mogą być stoły bilardowe wstawiane do barów — przedłużająca się partyjka nie pozwala odejść, a może przy okazji wzmagać pragnienie.

Ostatnio pomysł ten jest realizowany również z udziałem komputerów. Mniej więcej dwa lata temu klientom części barów w Stanach Zjednoczonych i Kanadzie zaoferowano nową grę o nazwie Trivia.

Właścicielem i organizatorem gry jest firma NTN Entertainment Network Steruje nią (grą!) komputer zainstalowany w siedzibie firmy w Carlsbad w Kalifornii. Uczestnicy, aktualnie z ok. 300 barów rozrzuconych po kontynencie, biorą udział w grze korzystając z linii telefonicznych. Sama gra ma charakter quizu — komputer zadaje pytania na najróżniejsze tematy, np. od problemów związanych z bólem zębów do prawa Ohma i gatunków drzew w Ameryce Południowej. Tematy i pytania są co wieczór nowe. Pytania wyświetlane są na zainstalowanym w barze monitorze, a uczestnicy wpisują odpowiedzi korzystając z przenośnych końcówek Typowa seria zawiera 15 pytań, wyniki podsumowywane są na bieżąco po każdym pytaniu. Po zakończeniu każdej serii na monitorze wyświetlane jest nazwisko zwycięzcy w barze, oraz wyniki uzyskane w innych barach

Scenariusz jest niezwykle prosty, ale okazuje się że rezultaty są bardzo dobre — oczywiście rezultaty mierzone wzrostem obrotów w barach które zakupiły Trivię (dla klientów gra zwykle jest bezpłatna). Może po prostu ludzie lubią konkurować, a nowa gra stwarza pole do rywalizacji. Zresztą rywalizacja ta już dawno wykroczyła poza pojedyncze bary. Rosnąca popularność Trivi doprowadziła do zorganizowania w marcu rozgrywek mistrzowskich (zwycięzył w nich bar Bridge Foy's z Filadelfii). Aby grę dodatkowo uatrakcyjnić, od czasu do czasu dla zwycięzcy fundowane są nagrody, np. kolorowy telewizor lub wycieczka na Hawaje.

Innym aspektem popularności gry jest prawdopodobnie to, że stwarza ona jakiś związek emocjonalny między zupełnie obcymi ludźmi, którzy do tej pory nie mieli ze sobą nic wspólnego (oprócz tego, że siedzieli w tym samym barze.).

Cóż jeszcze dodać, skoro w ostatecznym rozrachunku chodzi o interes — jeśli i sprzedawca gry, i jego klienci są zadowoleni — to chyba dobrze. Jeśli chodzi o mnie, to samo wymyślanie coraz to sprytniejszych sztuczek, aby przyciągnąć ludzi do siedzenia w barze, trudno mi ocenić jednoznacznie pozytywnie. Ale jedna rzecz robi mocne wrażenie w całej tej historii: poziom usług publicznej sieci telefonicznej, pozwalającej po prostu co wieczór realizować seanse łączności z komputerem z najzwyklejszych w świecie barów, porozrzucanych po, bądź co bądź, niemałym kontynencie.

(pan)



Za tymi sześcioma znaczkami kryje się niewielka płytka z jedenastoma układami scalonymi, pozwalająca na pięciokrotne przyspieszenie działania komputera wyposażonego w procesor 80386. Spośród trzech najważniejszych dla działania całego układu scalaków wyprodukowanych przez firmę Weitek, jeden służy do komunikowania się z głównym procesorem, a dwa pozostałe wykonują obliczenia. Podstawowe operacje zmiennoprzecinkowe wymagają 7 do 11 cykli zegara, co pozwala na osiągnięcie (przy uzyciu zegara 20 MHz) szybkości 2.8 miliona operacji zmiennoprzecinkowych na sekundę, przy prowadzeniu obliczeń na liczbach pojedynczej precyzji, czyli 32 bitowych (można również korzystać z podwojnej precyzji — 64 bity). Ponieważ zestaw rozkazów koprocesora mW1167 różni się od zestawu rozkazów 80387, użytkownik musi dysponować odpowiednimi kompilatorami, sprzedawanymi przez firmę MicroWay, będącą dystrubotorem opisanego urządzenia

Ktoś może zapytać, dlaczego na prezentowanym zdjęciu widać nie jedenaście, a dwanaście układów scalonych? Ten dwunasty to koprocesor 80387, z którego mogą korzystać stare programy Oba koprocesory mogą współistniec w jednym komputerze nie przeszkadzając sobie nawzajem.

Oprócz wersji podstawowej, nadającej się do zastosowania w komputerach serii PC, istnieje rownież wersja przeznaczona do systemu PS/2. Ceny w pierwszej połowie 1989 roku — od 1000 do 1600$. Cóż za szybkość trzeba płacić

(mb)

Kryptonim: mW1167

GEOS

Dla początkujących...

...to polskie tłumaczenie tytułu książki, jaką otrzymaliśmy dzięki uprzejmości firmy DATA BECKER. Jest to książka nie tyle dla początkujących, lecz raczej poradnik tłumaczący od podstaw zasady posługiwania się tym popularnym systemem.

Książka omawia kilka wersji systemu GEOS: Y1.2, Y1.3, GEOS 128 (znany pod nazwą GEOS Y1.4) oraz wersję niemieckojęzyczną przeznaczoną specjalnie do rozpowszechniania w RFN. Dzięki temu mogą z publikacji tej skorzystać użytkownicy Commodore 64, jak i 128 posiadający wersję anglo- i niemieckojęzyczną.

Jak wiadomo, GEOS doczekał się wielu programów współpracujących. Wspomniana książka omawia nie tylko podstawowe programy systemu, lecz również programy współpracujące, sprzedawane na oddzielnych dyskietkach. Oprócz wyczerpujących opisów programów wchodzących w skład systemu (GeoWrite, GeoPaint, Photo- i Text Manager, Preference Manager, Calculator, Notepad, Deks Top, Clock) autor książki omówił także: GeoDex (elektroniczny skorowidz), GeoPublish (program z serii Desk Top Publishing — projektowanie gazety), Icon Editor (program do zmiany wzorów ikon), Graphics Grabber (umożliwia przenoszenie grafiki z programów Newsroom, Print Master, Print Shop), GeoCalc (arkusz kalkulacyjny), FontPack (zestaw 20 czcionek), BLACK JACK (grę w „oczko") i kilka innych. W naszych polskich warunkach, gdy na giełdzie dostępny jest sam program, ale bez opisu, książka ta może okazać się bardzo przydatna dla wszystkich miłośników tego systemu.

Za szczególnie wartościowy uważam rozdział 5, w którym autor przedstawił kilka przykładów tworzenia tzw. listów okólnych (ta sama treść, lecz inny adresat) właśnie za pomocą programu GeoWrite Y2.0 i GeoDex. Z innych ciekawostek warto wymienić opis programu GeoCalc i zasady posługiwania się nim, tworzenie wykresów kołowych i słupkowych za pomocą GeoPaint, skalowanie rysunków, instalację i podstawowe zasady posługiwania się programem GeoPublish.

Na szczególne wyróżnienie zasługuje w tej publikacji dokładny opis wszystkich menu i opcji poszczególnych programów, co bardzo ułatwia pracę początkującym użytkownikom. Przy omawianiu wersji niemieckiej programu autor nie zapomniał o podaniu angielskich odpowiedników nazw zawartych w menu, dzięki czemu książka może służyć posiadaczom różnych wersji językowych systemu.

Pomijając szczególną wartość na polskim rynku, na którym chronicznie brakuje instrukcji obsługi do programów współpracujących z GEOS, publikacja ta może być bardzo pomocna podczas zapoznawania się z samym systemem (dotyczy to zwłaszcza początkujących). Duża ilość przystępnie omówionego materiału, sporo przykładów i rysunków oraz sam zakres książki, wykraczający sporo poza ramy jej tytułu sprawiają, iż będzie ona przydatną pozycją dla każdego zainteresowanego systemem GEOS.

Klaudiusz Dybowski

Manfred Tornsdorf GEOS FUER EINSTEIGER DATA BECKER GmbH Merowingerstr. 30, 4000 Dusseldorf Rok wydania: 1988, stron 252 Cena 29 DM ISBN 3-89011-275-7

SUPERSEJF 64

Przedstawiony poniżej program służy do zabezpieczania programów napisanych w BASIC-u. Jego główną zaletą jest przede wszystkim bardzo duża skuteczność zabezpieczenia.

Podany listing programu należy wpisać bez zmian (dwie pierwsze linie są niezbędne!). Po uruchomieniu programu do obszaru pamięci od adresu 2049 wpisywana jest automatycznie linia uruchamiająca (SYS 2061) oraz procedura dekodująca. Procedura ta jest następnie przenoszona w inny obszar pamięci, będący poza kontrolą interpretera (decyduje o tym zmienna SY, której przypisany jest normalnie adres 50000). W razie potrzeby adres ten można zmienić (procedura dekodująca jest relokowalna).

Po wykonaniu wszystkich procedur wstępnych na ekranie ukaże się instrukcja obsługi programu. Wczytaj teraz program do zakodowania i wpisz swoje hasło, poczynając od lewego górnego rogu ekranu (pozycja HOME). Hasło może zawierać do 256 znaków maksymalnie różnych od spacji (odstępu). Spacja powinna być ostatnim znakiem, natomiast wszystkie znaki zlokalizowane po niej nie będą już do klucza należały. Operacja kodowania programu nie nastąpi, jeśli w pamięci brak programu do kodowania, lub jeśli program ten jest już zakodowany. Po wpisaniu hasła wykonaj SYS ADRES, gdzie ADRES jest równy zmiennej SY (normalnie 50000).

Kodowanie polega na poddaniu operacji EXCLUSIYE OR (EXOR) zawartości wszystkich komórek pamięci, zajętych przez program z kolejnymi znakami hasła. Po osiągnięciu końca hasła pętla powtarza się aż do chwili osiągnięcia ostatniej komórki pamięci zajętej przez program, przy czym standardowe oznaczenie końca programu (trzy zera) nie jest kodowane. Zakończenie kodowania jest sygnalizowane przez wyświetlenie adresu końca programu w negatywie. Program (dłuższy o 90 bajtów) można teraz zapisać na taśmie lub dyskietce.

Dekodowanie programu jest w zasadzie automatyczne — wystarczy w tym celu wczytać program do pamięci, wpisać hasło (poczynając od lewego górnego rogu ekranu). Następnie wpisz RUN i wciśnij RETURN. Wpisywanie klucza powinno odbywać się przy standardowej konfiguracji pamięci ekranowej czyli wtedy gdy pamięć ekranowa rozpoczyna się od adresu 1024 ($0400). W obecności postronnych obserwatorów warto przed wpisaniem hasła zmienić kolor tekstu na kolor zgodny z tłem ekranu: w ten sposób hasło będzie niewidoczne.

Na koniec trochę statystyki. Wpisane hasło o długości 5 znaków daje nam liczbę kombinacji większą od biliona (255 ↑ 5 1E+12). Przy kluczu 16-znakowym liczba kombinacji przekracza możliwości obliczeniowe Commodore 64. Najlepsze efekty uzyskuje się przy kluczu o długości 5 do 10 znaków.

Marek Pomorski

```
DB   100 REM   UWAGA !!! PRZED URUCHOMIENIEM
C8   105 REM   ZAPISZ GO NA KASECIE LUB DYSKU
B7   110 :
A1   115 SY=50000:DIMX(73):FORK=1TO9
4F   120 X(48+K)=K:X(64+K)=K+9:NEXT
99   125 FORK=2049TOK+11:GOSUB190:POKEK,D
F2   130 NEXT:FORK=0TO88:GOSUB190:POKESY+K,D
0F   135 IFK<11THENNEXT
17   140 POKE2050+K,D:NEXT:FORK=SY+79TOK+23
7B   145 GOSUB190:POKEK,D:NEXT:READSK
8F   150 IFS<>SKTHENPRINT"POPRAW DANE !":END
CE   155 PRINTCHR$(147)CHR$(14)CHR$(154)
BD   160 PRINTCHR$(18)"↑ KLUCZ ↑ - SUPER";
69   165 PRINT" SEJF C-64 - M.POMORSKI"
AC   170 PRINT:PRINT
4B   175 PRINT" [1] WCZYTAJ PROGRAM":PRINT
68   180 PRINT" [2] WPISZ KLUCZ, WYKONAJ:"
88   185 PRINT:PRINT"SYS"SY:POKE43,91:NEW
84   190 READD$:L=X(ASC(RIGHT$(D$,1)))
3E   195 H=X(ASC(D$)):D=H*16+L:S=S+D:RETURN
0D   200 :
89   205 DATA 0B,08,C4,07,9E,32,30,36,31,00
B5   210 DATA 00,00,A5,2B,C9,01,F0,4F,AD,5C
B5   215 DATA 08,F0,4A,AD,00,04,C9,20,F0,43
3B   220 DATA 78,A9,5B,85,2B,85,FB,A9,08,85
B6   225 DATA FC,A5,2D,38,E9,03,85,FD,A5,2E
00   230 DATA E9,00,85,FE,A0,00,A2,FF,E8,BD
32   235 DATA 00,04,C9,20,D0,02,A2,FF,51,FB
79   240 DATA 91,FB,E6,FB,D0,02,E6,FC,A5,FE
BC   245 DATA C5,FC,D0,E6,A5,FB,C5,FD,D0,E0
87   250 DATA 58,20,59,A6,4C,AE,A7,6C,02,A0
39   255 DATA 00,A6,2D,A5,2E,D0,06,20,59,A6
43   260 DATA 6C,02,A0,C8,84,2B,E6,C7,20,CD
03   265 DATA BD,D0,F0,F0,EE, + 16613
```

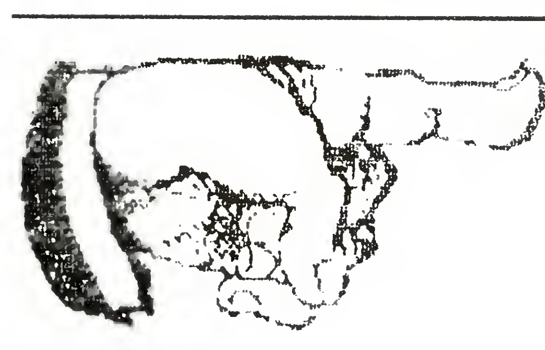
NIESPODZIANKA

Ten króciutki programik dla Commodore 64 demonstruje pewien efekt graficzny jakiego nie uzyskasz prawdopodobnie nigdy programując w BASIC. Po uruchomieniu programu możesz w zależności od własnych potrzeb zmienić kierunek przesuwu kasując linie 130 i 135 i usuwając instrukcje REM z linii 145 i 150. Program ten wykorzystuje przerwania IRQ i nie powinien pracować wraz z innym programem korzystającym z wektora zapisanego w komórkach $314 i $315 (odpowiednio 788 i 789).

Marcin Gruszczyk

```
99   100 FOR T=20000 TO 20053:READ A:POKE T,A:NEXT
D1   105 SYS 20000
B7   110 :
F0   115 DATA 120,169,045,141,020,003,169,078,141
         ,021,003
A1   120 DATA 088,096,162,000,160,000,142,032,208
         ,140,033
4F   125 DATA 208,232,200,224,255,192,255,208,242
         ,162,000
50   130 DATA 142,134,002,232,224,001,208,248,162
         ,077,142
4C   135 DATA 005,220,232,224,078,208,248,076,049
         ,234
F9   140 :
34   145 REM DATA 142,134,002,232,224,001,208,248
         ,162,076,142
CA   150 REM DATA 005,220,232,224,077,208,248,076
         ,049,234
```

PORADY SPOD LADY



Niedawno zauważyłem, że w moim C-64 nie działają pewne znaki w trzecim rzędzie (S,F,H,K i dalej do końca). [...] Gdy włączam joystick do 1 portu kursor cały czas jest w ruchu [...]"

Dariusz N., Kołobrzeg

Trudno mi ostatecznie rozstrzygnąć taką kwestię listownie, ale najprawdopodobniej ma Pan uszkodzony jeden z układów CIA (6526). Inne możliwości to: uszkodzenie mechaniczne przewodów łączących komputer z klawiaturą, zwarty joystick, ale obawiam się, że prawdziwa będzie moja pierwsza diagnoza.

●

Marek Cichosz

1. Tak, można połączyć Twój telewizor z komputerem za pomocą zwykłego przewodu antenowego, jaki otrzymałeś wraz z komputerem. Jeśli jednak Twój telewizor nie jest wyposażony w dekoder systemu PAL, to możesz mieć kłopoty z uzyskaniem dźwięku, nie uzyskasz także kolorów.
2. C-64 ma wejście dla myszki — po prostu przyłącza się ją do portu 1 zamiast joysticka.

●

1. Czy istnieje symulator lotu na C-16 lub PLUS/4?
2. Czy PLUS/4 jest całkowicie zgodny (BASIC) z C-16?
3. Jak uruchomić i korzystać z programów zapisanych w ROM PLUS/4?

Piotr Toma

1. [illegible] tak. Nie wiem jedynie, czy jest to pro[illegible] jak FLIGHT SIMULATOR II lub F-19 STE-[illegible] FIGHTER dla C-64.
2. BASIC Commodore PLUS 4 jest dokładnie taki sam jak [illegible] pomiędzy tymi komputerami polega przede wszystkim na większych możliwościach PLUS-4 wyposażonego w port użytkownika i programu zapisane w ROM.
3. Co do korzystania z programów zapisanych w ROM PLUS/4 to powinieneś zapoznać się z drugą częścią instrukcji obsługi komputera. Programy te można wywołać poprzez wciśnięcie klawisza funkcyjnego F1.

●

Po godzinie użytkowania mojego C-64 stwierdziłem, że mój komputer został wyposażony w uboższy procesor dźwiękowy. Po rozebraniu stwierdziłem, że jest to nowa wersja Commodore (opisywana przez pana J. Jasińskiego). Mój układ ma oznaczenie 8580 zamiast 6581 [...]"

Marek Mudy, Wałbrzych

Obawiam się raczej, że Pana komputer, a ściślej mówiąc układ 8580, jest po prostu uszkodzony. Firma Commodore raczej nie mogła zastąpić układu 6581 wersją zubożałą, gdyż natychmiast wieść ta rozniosłaby się po świecie i w efekcie nikt nie kupowałby nowej wersji C-64 niezgodnej z ponad 40000 programów dostępnych na rynku. Żelazną zasadą producentów (zarówno osprzętu jak i oprogramowania) jest zasada, iż nowa wersja tego samego komputera musi być stuprocentowo kompatybilna z wersjami starszymi. Co prawda zasada ta nie została wzięta pod uwagę przy produkcji pierwszych egzemplarzy AMIGI 1000, jednakże nic mi nie wiadomo o niekompatybilności poszczególnych wersji C-64.

●

Czy istnieje jakiś odpowiednik programowej instrukcji RESET (SYS 64738) dla komputerów C-16/116/PLUS4?"

J.K, Poznań

Owszem istnieje. Zaznaczam jednak, że nie jest to instrukcja, lecz konkretna procedura systemu operacyjnego. Dla Commodore 16/116/PLUS4 odpowiednikiem jest tu SYS 32793, dla Commodore 128 — SYS 65341. Należy pamiętać, że procedura ta kasuje program znajdujący się w chwili jej wykonania w pamięci.

SITO ERATOSTENESA
test szybkości

Jednym z najważniejszych programów wykorzystywanych do testowania języków programowania komputerów jest program generujący liczby pierwsze, oparty na algorytmie Eratostenesa.

Program ten zastosowałem do przeprowadzenia testu szybkości dla następujących sześciu translatorów: Action!, Kyan Pascal, Deep Blue C, Extendend fig—Forth, MMG Basic Compiler i Microsoft Basic.

Programujący w Kyan Pascalu z pewnością znają już ten program, figuruje on bowiem na dystrybucyjnej dyskietce pod nazwą PRIME. Listing 1 przedstawia tekst źródłowy programu w języku Action!. Teksty źródłowe programu w innych językach zamieszczone są w artykule Z. Szkaradnika [1]. Ten właśnie artykuł stanowił dla mnie inspirację do przeprowadzenia opisywanego testu szybkości. Z niego zaczerpnąłem wydruki programu, adaptując go do translatorów dostępnych dla Atari XL/XE.

Jak wynika z listingu 1, rozmiar deklarowanej tablicy FLAGS uniemożliwia przeprowadzenie tego testu dla Atari Basica i Turbo Basica XL. W przypadku tych translatorów tablica ta wymaga ponad 48 KB pamięci. Możliwe jest natomiast testowanie programu napisanego w Atari Basicu, skompilowanego przy użyciu kompilatora MMG Basic Compiler, dzięki opcji INTEGER kompilatora. Test ten można także przeprowadzić dla Microsoft Basica po zadeklarowaniu zmiennych typu całkowitego.

Wyniki testu zamieściłem w tabeli. Czas realizacji programów podany jest w cyklach zegara. Podałem także czasy względne, liczone względem Action! i Kyan Pascala.

Zgodnie z przewidywaniami najszybszym okazał się Action!. Program Action! wykonywany był pięć razy szybciej niż program w Kyan Pascalu. Wynik ten zachęcił mnie do przeprowadzenia porównania z komputerami kompatybilnymi z IBM PC/XT. Wykonałem więc analogiczny test szybkości (100 iteracji) dla Turbo Pascala 3.0 na mikrokomputerze Opus PC II. Rezultat przedstawia się następująco: 16 sekund (zegar 4.77 MHz) i 9 7 sekundy (zegar 8 MHz). Ten sam algorytm (100 iteracji) realizowany jest w Action! w czasie 21 sekund. A więc Action! jest naprawdę rewelacyjny, szczególnie gdy pamiętamy o tym, że mikroprocesor 6502 pracuje z zegarem o częstotliwości 1.77 MHz.

Czy najszybszy oznacza również najlepszy? Moim zdaniem, język Action! nie jest pozbawiony pewnych wad. Dyskowa wersja Action! pozostawia dla użytkownika tylko około 15—16 KB pamięci, w zależności od użytego dyskowego systemu operacyjnego. Drugą istotną wadą Action! jest niezbyt udana realizacja liczb zmiennoprzecinkowych, uniemożliwiająca zastosowanie ich w obliczeniach numerycznych. W związku z tym najwyższą moją ocenę uzyskał Kyan Pascal. Przyczyną takiej oceny jest nie tylko jego szybkość, lecz także inne powszechnie znane cechy Pascala, o których wiele pisano w licznych artykułach, zamieszczonych w „Bajtku" i „Komputerze".

Niekorzystnie wypadł wynik testu dla Deep Blue, C, co okazało się zgodne z wynikami otrzymanymi przez W. Zientarę [2]. Mała szybkość Deep Blue C, fatalna implementacja liczb zmiennoprzecinkowych oraz długi czas wymagany do uruchomienia programu, skutecznie zniechęcają do posługiwania się tym językiem programowania.

Duża jest natomiast szybkość realizacji skompilowanego programu, napisanego w Atari Basicu. Wynika stąd następujący wniosek dla programujących w tym języku: używajmy kompilatora, gdy nie musimy stosować liczb zmiennoprzecinkowych, w przeciwnym razie wybierzmy raczej Turbo Basic XL niż MMG Basic Compiler.

Translator	Czas realizacji programu	Porównanie względnych czasów realizacji	
Action!	107	1.00	0.19
Kyan Pascal	562	5.25	1.00
MMG Basic Compiler	839	7.84	1.49
Extended fig-Forth	881	8.23	1.57
Deep Blue C	3164	29.57	5.63
Microsoft Basic	13997	130.81	24.91

Translatorem, który zasługuje na większą uwagę, jest Extended fig—Forth, mimo że jest on o około 50% wolniejszy od Kyan—Pascala Listing 2 przedstawia tekst programu w tym języku. Forth posługuje się w obliczeniach strukturą stosu i odwrotną notacją polską, co stanowi dużą trudność dla uczących się tego języka. Lecz posiada także niewątpliwe zalety. Extended fig—Forth pozostawia do dyspozycji użytkownika około 36 KB pamięci. Jest on równocześnie językiem pamięciooszczędnym, tworzącym wyjątkowo zwięzły kod. Będąc językiem strukturalnym wysokiego poziomu, dysponuje równocześnie szeregiem możliwości zastrzeżonych dla asemblera. Jedną z bardziej istotnych jego zalet jest konwersacyjny tryb pracy. Zachęcam redaktorów Klanu Atari do przedstawienia tego mało znanego w naszym kraju języka Czytelnikom „Bajtka".

LITERATURA

1. Z. Szkaradnik, Porównanie języków programowania mikrokomputerów, INFORMATYKA, nr 11—12, 1985.

2. W. Zientara, Action! znaczy szybkość, BAJTEK, nr 4, 1988.

Mieczysław Gabryszewski

LISTING 1

```
;SITO ERATOSTENESA

PROC PRIME()
 BYTE ARRAY FLAGS(8190)
 BYTE TM1=19,TM2=20
 INT I,K,PRIME,COUNT,TIME
 INT SIZE=[8190]

 TM1=0 TM2=0
 PUTE() PRINTE("1 iteracja")
 COUNT=0
 FOR I=0 TO SIZE
  DO
   FLAGS(I)=1
  OD
 FOR I=0 TO SIZE
  DO
   IF FLAGS(I) THEN
    PRIME=I+I+3
    ;PRINTC(PRIME) PUTE()
    K=I+PRIME
    WHILE K<=SIZE
     DO
      FLAGS(K)=0
      K==+PRIME
     OD
    COUNT==+1
   FI
  OD
 PRINTC(COUNT) PRINTE(" liczb pier
wszych")
 TIME=TM1*256+TM2
 PRINTC(TIME) PRINTE(" cykli")

RETURN
```

LISTING 2

```
( SITO ERATOSTENESA )

8190 CONSTANT SIZE
0 VARIABLE FLAGS
SIZE ALLOT

: PRIME
CR ." 1 iteracja" CR
FLAGS SIZE 1 FILL
0 SIZE 0 DO
 FLAGS I + C@
 IF I DUP + 3 + DUP I +
  BEGIN DUP SIZE <
  WHILE 0 OVER FLAGS + C! OVER +
  REPEAT DROP DROP 1+
 THEN
LOOP
. SPACE ." liczb pierwszych"
CR ;

: TEST
0 19 C! 0 20 C! PRIME
19 C@ 256 * 20 C@ +
. SPACE ." cykli"
CR ;
```

MULTI—KOPIER

Wszyscy posiadacze jednej stacji dysków wiedzą zapewne jak uciążliwe jest kopiowanie większej ilości krótkich programów przy pomocy DOS-u.

Częste przekładanie i zamienianie dyskietek jest bardzo niewygodne i przeważnie łączy się z wieloma pomyłkami, często nieprzyjemnymi w skutkach. Z problemem tym nie sposób się nie zetknąć przy tworzeniu dyskietek roboczych takich programów jak Deep Blue C czy Kyan Pascal, lub przy okazji robienia porządków we własnych dziełach.

Kłopot ten będziesz miał za sobą, Drogi Czytelniku, jeżeli poświęcisz parę godzin na wpisanie tego programu. Dobry program kopiujący powinien posiadać następujące zalety: niewielka długość, dużo wolnej pamięci, wygoda obsługi, użyteczność, uniwersalność. Sądzę, że nie będzie w tym wiele przesady jeżeli powiem, że Multikopier spełnia te wymagania. Posiada on około 45 kilobajtów wolnej pamięci (wielkość ta zależy od wersji DOS-u, z którym współpracuje). Taka wielkość wolnej pamięci umożliwia kopiowanie nie tylko wielu krótkich programów, ale także możliwe jest kopiowanie pojedynczych programów tak długich jak Amaurote lub Starquake. Multikopier pozwala również na jednoczesne wczytanie 30 programów (plików). Nazwy zbiorów znajdujących się w pamięci uwidocznione są na ekranie. Multikopier współpracuje ze wszystkimi typami stacji dysków oraz ze wszystkimi DOS-ami.

Mój program jest na tyle prosty w obsłudze, że ograniczę się tylko do wyjaśnienia najistotniejszych kwestii. **Opcja 1** — ładowanie pliku — po jej wywołaniu program poprosi o podanie nazwy, może ona zawierać rozszerzenia takie jak „?" lub „*". Następnie program podejmie próbę wczytania wszystkich plików o podanej nazwie. Na pytanie „czy kopiować" można udzielić następujących odpowiedzi: **T** — ładuj, **N** — nie ładuj, **W** — wyjdź z opcji ładowania. Wszystkie błędy, jakie wystąpiły w trakcie wczytywania, zostaną zasygnalizowane.

W przypadku **opcji 2** — zapis — najpierw podajemy numer zapisywanego pliku według numeracji na ekranie, następnie należy podać nazwę. Jeśli zamiast tego naciśniemy ‹**RETURN**›, zbiór zostanie zapisany pod nazwą, z jaką został wczytany. Opcje 3 i 4 nie wymagają komentarza.

Teraz kilka słów o budowie programu. Duża ilość wolnej pamięci wynika z użycia podanej do **SPEEDTRANS** procedury maszynowej, umożliwiającej korzystanie z dodatkowej pamięci RAM, niedostępnej z poziomu BASICA, programowo podłączanej na miejscu systemu operacyjnego. Zapis i odczyt realizowany jest również przez procedurę maszynową współpracującą z BASIC-em. W czasie operacji I/O ekran jest wygaszany, co znacznie skraca długość przerw przy zapisie lub odczycie sektorów. (Program w BASIC-u współpracujący z procedurą maszynową działa szybciej).

Podawana na ekranie liczba wolnych bajtów pamięci nie jest wartością dokładną. Dokładne podanie tej wielkości przy zachowaniu uniwersalności programu kopiującego nie jest rzeczą prostą, stąd podaje ona tylko szacunkowo tą wartość.

Multikopier został napisany z troską, aby zajmował jak najmniejszy obszar pamięci, zwracam uwagę na zastąpienie często używanych liczb zmiennymi L0, L1, L2 itp. Mimo to tego około 95 — wierszowego programu nie zaliczyłbym do najkrótszych. Proponuję więc nie wpisywać komentarzy (REM).

Na zakończenie życzę dużo cierpliwości przy wpisywaniu i wiele satysfakcji z jego wykorzystania.

Andrzej Zalewski

LISTING 1

```
        0100        *=  $0600
        0110 ;
        0120        RTI
        0130        PLA
        0140        PLA
        0150        STA SKAD+2
        0160        PLA
        0170        STA SKAD+1
        0180        PLA
        0190        STA GDZIE+2
        0200        PLA
        0210        STA GDZIE+1
        0220 ; ustawienie adresow
        0230        SEI
        0240        LDA #$FE
        0250        STA 54017
        0260        LDA #0
        0270        STA $FFFA
        0280        STA $FFFE
        0290        LDA #6
        0300        STA $FFFB
        0310        STA $FFFF
        0320 ; przeniesienie
        0330        LDX #0
        0340 SKAD   LDA $8000,X
        0350 GDZIE  STA $8000,X
        0360        INX
        0370        BNE SKAD
        0380 ; normalne parametry, powrot
        0390        CLI
        0400        LDA #253
        0410        STA 54017
        0420        RTS
        0430 ; odczyt 256 bajtow (IOCB1)
        0440        PLA
        0450        LDX #$10
        0460        PLA
        0470        PLA
        0480        STA 850
        0490        PLA
        0500        STA 853
        0510        PLA
        0520        STA 852
        0530        LDA #0
        0540        STA 856
        0550        LDA #1
        0560        STA 857
        0570        JMP $E456
```

LISTING 2

```
CT   0 REM MULTI KOPIER
LT   1 REM Andrzej Zalewski
UP   2 REM Copyright (c) Bajtek
NI   3 REM
US   10 L1=1:L2=L1+L1:L4=L2+L2:L100=100:START=90:NORM=2000:ANORM=NORM+L100:STR=256:L17=17:STAT=851:L3=L1+L2
JZ   20 DIM A$(L17),B$(L17),N$(L17),F$(510),F(30,2)
AO   30 FILE=L1:SCR=559:L34=34:L15=15
IK   60 FOR N=1536 TO 1622:READ A:POKE N,A:S=S+A:NEXT N
RA   70 OPEN #L2,L4,L0,"K":POKE 703,24:POKE 752,L1
TN   80 POKE 82,L0:? "K    MULTI KOPIER   by  Andrzej Zalewski      1=load 2=save 3=dir 4=new free:"
YW   85 WOLNE=60160-PEEK(144)-PEEK(145)*STR:GOSUB 2300:BUF=L100+PEEK(144)+256*PEEK(145):TOP=BUF+L4*L100
QL   90 TRAP START:GET #L2,KEY:ON KEY-48 GOSUB 200,500,L100,160:GOTO START
BW   98 REM
UM   99 REM    KATALOG
WD   100 TRAP 150:GOSUB ANORM:OPEN #L1,6,L0,"D:*.*"
HV   120 FOR M=L1 TO L4-L1
NV   130 INPUT #L1;A$:? A$,:INPUT #L1;A$:? A$
MQ   140 NEXT M:GET #L2,KEY:GOTO 120
WQ   150 POP :CLOSE #L1:GET #L2,KEY:GOSUB NORM:RETURN
RU   158 REM
LC   159 REM    NEW
ME   160 GOSUB ANORM:? "NA PEWNO? [T/N]"
BL   170 GET #L2,KEY
UO   180 IF KEY=84 THEN RUN
JP   190 GOSUB NORM:RETURN
RZ   197 REM
NU   198 REM WCZYTAJ PROGRAMY
SF   199 REM
KK   200 GOSUB ANORM:M=L0:? "Podaj nazwe pliku :":INPUT A$:B$="D:":B$(L3,L17)=A$
UQ   209 REM CAT,ODCZYTAJ NAST.PROGRAM
ZW   210 M=M+L1:IF FILE>L15+L15 THEN 360
MU   215 TRAP 217:OPEN #L3,L3+L3,L0,B$:GOTO 220
ME   217 ? "Blad nr ";PEEK(195):GET #L2,KEY:GOTO 360
QA   220 FOR N=L1 TO M:INPUT #L3;A$:NEXT N:CLOSE #L3:Q=(FILE-L1)*L17:F$(Q+L1,Q+L17)=A$
ES   225 IF A$(L4,L17-L1)=" FREE SECTORS" THEN 360
AO   230 ? "CZY WCZYTAC [T/N/W]  :";A$:INPUT N$:IF N$="N" THEN 210
XD   235 IF N$="W" THEN 360
SY   240 GOSUB 2400
WB   250 POKE SCR,L0:TRAP 275:OPEN #L1,L4,L0,N$:AD=TOP:DL=L0
IF   260 X=USR(1594,L4+L3,BUF):X=USR(1537,BUF,AD)
CE   270 N=PEEK(856)+STR*PEEK(857):DL=DL+N:AD=AD+N:GOSUB 2500
JS   275 N=PEEK(STAT)
XW   280 IF N=L1 THEN 260
RK   290 IF N=136 THEN 320:REM POTWIERDZ WCZYTANIE
IL   300 CLOSE #L1:POKE SCR,L34:POKE 752,L1:? "Blad nr ";N
JO   305 ? "Powtorzyc-P, dalej-D, akceptuj-A";:INPUT A$:IF A$="P" THEN 250
FR   306 IF A$="A" THEN 320
MF   310 GOTO 210
ZY   320 F(FILE,L1)=TOP:F(FILE,L2)=DL:GOSUB 2500:TOP=AD:WOLNE=WOLNE-DL:GOSUB NORM:GOSUB 2300
QM   330 POKE SCR,L34:POSITION L0+(FILE>L15)*20,L3+(FILE<=L15)*FILE+(FILE>L15)*(FILE-L15)
PD   340 ? FILE;".";F$(Q+L3,Q+L17):GOSUB ANORM
EX   350 FILE=FILE+L1:CLOSE #L1:IF FILE>L15+L15 THEN 360
NC   355 GOTO 210
ML   360 POP :CLOSE #L1:CLOSE #L3:? "K":GOSUB NORM:GOTO START
RW   495 REM
YD   496 REM  PROCEDURA ZAPISU
SC   497 REM
SF   498 REM
XQ   500 IF FILE=L1 THEN RETURN
MI   502 TRAP 610:GOSUB ANORM:? "Podaj nr pliku do zapisu";:INPUT M:IF M>(FILE-L1) OR M<L1 THEN 500
YL   505 M=INT(M):Q=(M-L1)*L17
SB   510 ? "Podaj nazwe pliku:";:INPUT A$:IF A$="" THEN A$=F$(Q+L1,Q+L17):GOSUB 2400:GOTO 530
QN   520 N$="D:":N$(L3,L17)=A$
AB   530 POKE SCR,L0:TRAP 580:OPEN #L1,L4+L4,L0,N$
US   540 DL=F(M,L2):AD=F(M,L1)
NG   550 X=USR(1537,AD,BUF)
ID   555 IF DL<STR OR DL=L0 THEN FOR M=L1 TO DL:PUT #L1,PEEK(BUF+M-L1):NEXT M:GOTO 600:REM QUIT
SL   560 X=USR(1594,11,BUF)
NS   570 DL=DL-STR:AD=AD+STR:GOSUB 2500
OT   580 M=PEEK(STAT):IF M=L1 THEN 550
UK   590 POKE SCR,L34:? "Blad nr ";M:GET #L2,KEY
CU   600 POKE SCR,L34:CLOSE #L1:? "K":GOTO 500
VE   610 GOSUB NORM:GOTO START
HI   1997 END
ZG   1998 REM    PROCEDURY POMOCNICZE
ZQ   1999 REM    NORMALNIE
JO   2000 POKE 752,L1:? "K":POKE 703,24:RETURN
MY   2099 REM  ANORMALNIE
JU   2100 POKE 752,L0:POKE 703,L4:? "K":RETURN
KG   2198 REM
NT   2199 REM  DANE DLA PROCEDURY
UL   2200 DATA 64,104,104,141,44,6,104,141,43,6,104,141,47,6,104,141,46,6,120,169,254,141,1,211,169,0,141
XF   2210 DATA 250,255,141,254,255,169,6,141,251,255,141,255,255,162,0,189,120,230,157,64,156,232,208,247,88,169
GY   2220 DATA 253,141,1,211,96,104,162,16,104,104,141,82,3,104,141,85,3
QA   2230 DATA 104,141,84,3,169,0,141,88,3,169,1,141,89,3,76,86,228
PU   2299 REM  DRUKUJ WOLNA PAMIEC
UY   2300 POSITION 31,L1:? WOLNE;"     ":RETURN
MT   2399 REM  UTWORZ NAZWE PROGRAMU
LC   2400 N$="D:????????.???"
BA   2410 FOR N=3 TO 10:IF A$(N,N)=" " THEN POP :GOTO 2430:REM OUT
WX   2415 IF A$(N,N)="_" THEN 2425
ZZ   2420 N$(N,N)=A$(N,N)
IB   2425 NEXT N
ET   2430 N$(12,14)=A$(11,13)
AU   2440 RETURN
SZ   2499 REM  PRZESKOCZ ZAKAZANE OBSZARY
PI   2500 IF AD>39712 AND AD<40960 THEN AD=40960
MD   2510 IF AD>52992 AND AD<57344 THEN AD=57344:REM TA WARTOSC MOZE BYC MNIEJSZA O OK.2000
RN   2520 IF AD>65200 THEN ? "BRAK PAMIECI RAM":POP :POP :POKE SCR,L34:GET #L2,KEY:GOTO 360
AT   2530 RETURN
```

KLAN SPECTRUM

SCREEN COMPRESSOR PLUS

Jak wiadomo, pamięć obrazu Spectrum ma 6.912 bajtów. Tak więc programiści często stoją przed dylematem: czy poświęcić blisko 7 KB na jeszcze jeden obrazek, kosztem programu, czy też z tego zrezygnować. 7 KB to niemało, zmieszczą się tam np. 3 muzyczki na generator AY. Najczęściej więc jedynym zapamiętanym w tej postaci obrazkiem jest screen, umieszczany zwykle jako drugi blok programu na taśmie.

Wczytywanie trwa te kilkanaście sekund, oprócz tego zajmuje się kawałek taśmy. Zaś przy kopiowaniu programu na dyskietkę rezygnuje się zazwyczaj z obrazka.

Jeśli jednak musi on być, to co zrobić, by zajmował jak najmniej? Odkurzamy stare kasety i znajdujemy program **Compressor.** Postępujemy według instrukcji i po chwili screen zajmuje już ok. 4 KB. Ale radość trwa krótko. Okazuje się, że dekompresja trwa przeszło sekundę! To za długo, by zalety **Compressora** mogły być wykorzystane w naszym programie.

Sięgamy więc po rodzime czasopisma komputerowe gdzie podobnych programów ukazało się już kilka. Ale i tu zawód — dekompresja jest co prawda błyskawiczna, lecz obrazek po kompresji zajmuje niewiele mniej miejsca, niż przed, czasem nawet więcej!

Co zrobić w takiej sytuacji? Wystarczy „wklepać" przedstawiony obok program **„Screen Compressor Plus"**, w postaci dwu listingów. **SCP** łączy zalety przedstawionych wcześniej programów, jest przy tym pozbawiony ich wad.

Kompresja przy pomocy programu **SCP** odbywa się czterema metodami, przy czym wybierana jest metoda najoszczędniejsza. Ogólna zasada jest następująca: pamięć obrazu przebiegana jest porcjami po 8 bajtów lub bajt po bajcie w celu znalezienia najczęściej występującej wartości. Odpowiednie odcinki pamięci obrazu przekodowywane są i umieszczane dalej. Krótka procedura jest w stanie odkodować treść obrazu i umieścić ją w odpowiednim miejscu.

Aby otrzymać gotowy **SCP**, należy wpisać program z listingu 1 i nagrać na taśmę przez **SAVE "SCP"LINE 9999.** Następnie wpisać na czystą pamięć listing 2 i uruchomić. Ewentualne błędy zostaną zasygnalizowane, zaś bezbłędne wpisanie zakończy się komunikatem „Start tape...". Należy nagrać blok kodu na taśmę i zweryfikować. Program **Screen Compressor** jest gotowy do użytku.

Wczytywane przez program screeny powinny być bez nagłówka. Jeśli więc on występuje, należy przewinąć taśmę za nagłówek i włączyć rozpoczynając odczyt bloku bajtów.

Nagrany na taśmę skompresowany obrazek wczytuje się i wywołuje na ekran następująco:

CLEAR 49999: LOAD ""CODE 50000: RANDOMIZE USR 50000: PAUSE 0.

Życzę wszystkim, którzy będą korzystać z SCP, owocnej pracy i... dobrej kompresji.

Tomasz J. Puwalski

```
  10 GO SUB 1000: PRINT AT 12,0;"Wczytywanie obrazka bez naglowka": RANDOMIZE USR 40000+USR 40309
  20 GO SUB 1000: PRINT  FLASH 1'" KOMPRESJA "''
  30 LET d=9999
  40 FOR n=3 TO 6
  50 GO SUB n*100: LET l=m-50000: PRINT l: IF l<d THEN  LET d=l: LET o=n
  60 NEXT n
  70 PRINT '"WYBRANA ";: GO SUB o*100: PRINT d
  80 GO SUB 1000: PRINT ''"Dlug. po kompresji = ";d
  85 PRINT ' INVERSE 1'" 1 "; INVERSE 0;" Zapis na tasmie"' INVERSE 1'" 2 "; INVERSE 0;" Pokazanie obrazka"' INVERSE 1'" 3 "; INVERSE 0;" Wczytanie nowego obrazka"
  90 IF INKEY$="1" THEN  GO SUB 2000: GO TO 80
  95 IF INKEY$="2" THEN  RANDOMIZE USR 50000: PAUSE 1: PAUSE 0: GO TO 80
 100 IF INKEY$="3" THEN  GO TO 10
 105 GO TO 90
 300 PRINT "METODA 1 - dlug. ";: RANDOMIZE USR 40375: LET m=USR 40149: RETURN
 400 PRINT "METODA 2 - dlug. ";: RANDOMIZE USR 40383: LET m=USR 40024: RETURN
 500 PRINT "METODA 3 - dlug. ";: RANDOMIZE USR 40391: LET m=USR 40067: RETURN
 600 PRINT "METODA 4 - dlug. ";: RANDOMIZE USR 40411: LET m=USR 40121: RETURN
1000 CLS : PRINT AT 1,5;"SCREEN COMPRESSOR PLUS";AT 2,5;"Bajtek & MIKROPOL 1989"''" Tomasz J. Puwalski  Bydgoszcz": RETURN
2000 INPUT AT 0,0;"Podaj nazwe (max.10 liter)";AT 1,0; LINE a$
2010 IF NOT LEN a$ OR LEN a$>10 THEN  GO TO 2000
2020 SAVE a$CODE 50000,d: RETURN
9998 LOAD ""CODE 50000: RANDOMIZE USR 50000: RANDOMIZE USR 40734: GO TO 20: REM Re-compression
9999 PAPER 0: BORDER 0: INK 7: CLEAR 39999: LOAD ""CODE : RUN
```

```
   5 CLEAR 39999
  10 FOR n=0 TO 33: LET s=0
  20 FOR m=0 TO 21
  30 READ a: POKE 40000+22*n+m,a: LET s=s+a
  40 NEXT m: READ s1: IF s<>s1 THEN  PRINT "Popraw dane w linii ";1000+n*10: STOP
  50 NEXT n
  60 SAVE "COMPR.+ c"CODE 4e4,746
  70 VERIFY "COMPR.+ c"CODE
  80 RANDOMIZE USR 0
1000 DATA 221,33,0,64,221,229,17,0,27,213,62,255,55,205,86,5,193,225,17,0,229,195,2552
1010 DATA 195,51,243,217,1,0,0,221,33,182,195,217,33,0,229,22,3,229,14,0,229,6,2320
1020 DATA 8,205,37,157,36,16,250,225,35,13,32,242,225,1,0,8,9,21,32,231,195,10,1988
1030 DATA 157,243,217,1,0,0,221,33,193,195,217,33,0,229,22,3,229,30,8,229,14,8,2282
1040 DATA 229,6,32,205,37,157,35,16,250,225,36,13,32,242,225,1,32,0,9,29,32,231,2074
1050 DATA 225,1,0,8,9,21,32,220,195,10,157,243,217,1,0,0,221,33,133,195,217,33,2171
1060 DATA 0,229,1,0,24,205,37,157,35,11,120,177,32,247,195,10,157,243,217,1,0,0,2098
1070 DATA 221,33,195,195,217,33,0,229,22,32,229,30,3,213,229,14,8,229,6,8,205,37,2388
1080 DATA 157,36,16,250,225,17,32,0,25,13,32,239,225,17,0,8,25,209,29,32,226,225,2038
1090 DATA 35,21,32,218,33,0,253,1,0,3,205,37,157,35,11,120,177,32,247,217,175,184,2193
1100 DATA 196,103,157,217,221,229,193,251,201,126,217,87,205,45,157,217,201,120,254,0,32,45,3474
1110 DATA 62,154,186,40,33,121,254,2,48,7,12,221,114,0,221,35,201,221,126,254,221,190,2723
1120 DATA 255,32,242,186,32,239,6,3,221,54,254,154,221,119,255,201,221,35,221,35,4,24,3014
1130 DATA 241,122,221,190,255,32,11,4,192,221,112,0,1,0,0,221,35,201,205,103,157,24,2548
1140 DATA 189,243,33,80,195,175,71,119,35,16,252,229,17,0,229,26,79,33,80,195,9,62,2367
1150 DATA 91,190,56,1,52,19,186,32,240,225,29,43,86,186,56,2,122,75,29,16.246,121,2103
1160 DATA 50,51,157,50,83,157,50,15,158,50,68,158,50,126,158,50,195,158,50,8,159,251,2252
1170 DATA 201,33,231,157,1,79,0,24,14,33,90,158,1,66,0,24,6,33,156,158,1,77,1543
1180 DATA 0,17,80,195,237,176,33,54,158,',36,0,195,195,51,33,233,158,17,80,195,1,2145
1190 DATA 53,0,195,195,51,243,205,146,23,59,59,225,17,111,0,25,229,217,209,6,0,217,2485
1200 DATA 33,0,64,22,32,229,30,3,213,229,14,8,229,6,8,217,120,254,0,32,12,26,1781
1210 DATA 254,154,19,32,8,26,79,19,26,71,19,5,121,217,119,36,16,231,225,17,32,0,1726
1220 DATA 25,13,32,220,225,17,0,8,25,209,29,32,207,225,35,21,32,199,33,0,88,1,1676
1230 DATA 0,3,217,120,254,0,32,12,26,254,154,19,32,8,26,79,19,26,71,19,5,121,1497
1240 DATA 217,119,35,11,120,177,32,228,251,201,243,205,146,23,59,59,225,17,98,0,25,229,2720
1250 DATA 217,209,6,0,217,33,0,64,22,3,229,14,0,229,6,8,217,120,254,0,32,12,1892
1260 DATA 26,254,154,19,32,8,26,79,19,26,71,19,5,121,217,119,36,16,231,225,35,13,1751
1270 DATA 32,223,225,1,0,8,9,21,32,212,243,205,146,23,59,59,225,17,109,0,25,229,2103
1280 DATA 217,209,6,0,217,33,0,64,22,3,229,30,8,229,14,8,229,6,32,217,120,254,2147
1290 DATA 0,32,12,26,254,154,19,32,8,26,79,19,26,71,19,5,121,217,119,35,16,231,1521
1300 DATA 225,36,13,32,223,225,1,32,0,9,29,32,212,225,1;0,8,9,21,32,201,243,1809
1310 DATA 205,146,23,59,59,225,17,49,0,25,229,217,209,6,0,217,33,0,64,1,0,27,1811
1320 DATA 217,120,254,0,32,12,26,254,154,19,32,8,26,79,19,26,71,19,5,121,217,119,1830
1330 DATA 35,11,120,177,32,228,251,201,33,0,64,17,0,229,1,0,27,195,195,51,0,0,1867
```

MÓZGPROCESOR

– polska gra tekstowa

Gry tekstowe (adventure) są najprzyjemniejszą formą spędzania czasu, oczywiście dla światłych Czytelników „Bajtka". Pożądana jest jednak dobra znajomość angielskiego, gdyż w tym języku najczęściej rozgrywa się akcja gry.

Mamy jednak nie gęsi, lecz swój język i preferujemy używanie jego. Dlatego z lepszym lub gorszym skutkiem powstają i polskie gry tekstowe, pisane — nie ma chyba wątpliwości — przez Polaków.

Krom-Kruak, Puszka Pandory, Smok Wawelski to znane gry. Wielu woli je niż np. Gremlins, Sherlock Holmes, gdyż są po polsku. I nie są to żadne przeróbki, lecz programy pisane od zera.

Brakuje im jednak tego wygładzenia, tego smaku, który zwykle towarzyszy „oryginalnej", zachodniej grze. Dlaczego? Nad grą pracują zastępy ludzi, którzy dysponują sprzętem PC i niezliczoną ilością gotowych procedur. Gry pisane są w języku strukturalnym i dopiero potem kompilowane i przekodowywane na Spectrum czy inny mikrokomputer. Dlatego nawet najzdolniejszy programista, gdy jest sam, nie dokona tego, co pięciu „średniaków" razem i z dobrym sprzętem.

Są jednak trzej ludzie, którzy postanowili napisać polską grę tekstową z prawdziwego zdarzenia. Założyli w tym (i nie tylko) celu firmę — Computer Adventure Studio i po dwuletniej pracy stworzyli arcydzieło — Mózgprocesor, którego wersja powstaje i na IBM.

Piotr Kucharski, Wiesław Florek i Krzysztof Piwowarczyk z Bochni, autorzy i Smoka Wawelskiego dobrze wiedzą, co to znaczy „programować". Ich gra zajmuje całą pamięć Spectrum i nie ma w niej ani jednego zbędnego bajtu. Oprawa graficzna i dźwiękowa mogą budzić zachwyt i czyniły to podczas testowania programu.

Gra traktuje o poszukiwaniach tytułowego Mózgprocesora na Wyspie Thompsona. Potrzebny jest on, by uratować światowej sławy profesora, który po sześciu latach pracy zbudował komputer biologiczny, naśladujący prace ludzkiego mózgu. W drodze na konferencję profesor uległ wypadkowi i uratować go może jedynie wszczepienie własnego wynalazku.

Na poszukiwania wyrusza doskonale wyszkolony agent, który ma cztery godziny na spenetrowanie wyspy. Wyposażony został w zegar z kompasem i licznikiem Geigera-Mullera. Wszystko spoczywa więc w jego rękach.

Program gry napisany jest konsekwentnie. Bohater nie może więc nieść naraz zbyt wielu lub zbyt ciężkich przedmiotów, męczy się, ładunek przeszkadza mu, ciało jego narażone jest na uszkodzenia.

Wydawanie komend i otrzymywanie odpowiedzi odbywa się w języku polskim. Polskie znaki uzyskuje się przez wciśnięcie odpowiedniej litery wraz z Symbol Shift. Wyjątkiem jest Ż — Symbol Shift i X.

W czasie gry ekran podzielony jest na cztery części. Obejmują one: kolorowy rysunek miejsca, w którym znajduje się bohater, wspomniany zegar z kompasem i licznikiem G-M (+ wskaźnik napromieniowania), okno ze słownym opisem miejsca oraz okno dialogowe, służące do wprowadzania komend.

Jeszcze jedną rzeczą, która wyróżnia Mózgprocesor spośród innych gier, jest niesłychanie bogaty słownik i bardzo rozbudowany procesor komend. Mózgprocesor pod tym względem stoi wyżej od Hobbita i niewiele niżej od Police Quest na IBM.

Jedyna, drobna usterka, to oznaczenie kierunków geograficznych. Według mnie nie powinno zostać to spolszczone, gdyż tymczasem „północ", „południe", „wschód" i „zachód", obok tego, że rozpoznawane są w całości, kodowane są jako „pn", „pd", „ws" i „zd". Czy nie lepiej pozostać przy N, S, E, W?

Czy dwuletnia praca firmy CAS miała pójść na marne? Dorzucić jeszcze jedną złotówkę do kiesy giełdowych handlarzy. Czy też być idealistycznym powodem dumy programistów? Na pewno nie. Gra Mózgprocesor jest więc rozprowadzana przez firmę CAS na taśmach. Wraz z estetycznie opracowaną i wydrukowaną instrukcją jest na pewno poważną propozycją dla graczy. Hackerzy znajdą w niej też smakowity kąsek; gra została zabezpieczona przed wszystkimi programowymi kopierami, dodatkowo każda kopia posiada numer ewidencyjny. Na pudełku ani okładkach kasety nie ma bzdurnych napisów „kopiowanie zabronione", w loaderze zawarte jest jednak ostrzeżenie „za nielegalne rozpowszechnianie tej kopii grozi kara".

Marcin Przasnyski



OŚMIORNICE



Gra polega na sterowaniu łodzią podwodną, która w krótkim czasie (wskaźnik w lewym górnym rogu ekranu) musi wyłowić pięciu nurków (ukazują się oni w prawym górnym rogu). Utrudnienie stanowią pływające w pobliżu ośmiornice i skaliste dno, o które możesz się rozbić. Kierowanie łodzią: klawisze Q, A, O, P.

Zaczynaj więc!

Michał Kunze

```
  10 REM Ośmiornice  1C
  40 FOR f=USR "a" TO USR "d"+7  DF
  50 READ a: POKE f,a  A3
  60 NEXT f  8C
  65 POKE USR "s",16  70
  70 DATA 0,12,158,255,62,0,0,0,93,42,255,60,74,82,146,74,0,24,24,16,124,16,40,68,170,85,170,85,170,85,170,85  78
  80 PAPER 7: INK 0: BORDER 7: CLS  6A
  90 FOR f=1 TO 5: PAUSE 30: READ a,b,a$  DD
 100 FOR s=1 TO LEN a$: PRINT AT a,b;a$(1 TO s): BEEP .02,-20: NEXT s: NEXT f  58
 110 DATA 3,7,"Bajtek i M-Software",4,10,"przedstawiaja:",6,11,"OSMIORNICE",8,11,"autorstwa",9,12,"M.KUNZE"  67
 120 FOR r=6 TO 26 STEP 2: PRINT AT 2,r; INK RND*5;"■";AT 10,r; INK RND*5;"■": NEXT r  EE
 130 FOR r=2 TO 10 STEP 2: PRINT AT r,6; INK RND*5;"■";AT r,26; INK RND*5;"■": NEXT r  F8
 140 FOR f=6 TO 0 STEP -1: PRINT AT 12,10; INK f;"Ucisnij "S"": IF INKEY$<>"s" THEN PAUSE 5: NEXT f: PAUSE 5: GO TO 140  0B
 160 REM Start!  C5
 170 PAPER 5: INK 0: BORDER 5: CLS  DC
 180 FOR f=0 TO 31: READ w: FOR g=21 TO w STEP -1: PRINT AT g,f;"■": NEXT g: NEXT f  6C
 190 DATA 17,16,15,16,14,13,12,15,12,11,10,11,12,13,14,15,17,16,15,14,13,12,13,15,14,15,16,17,18,21,14,15  4D
 200 PRINT AT 0,0;"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"  49
 210 READ a$: FOR f=1 TO LEN a$: PRINT AT 0,11;a$(1 TO f): BEEP .25,-25: NEXT f  FF
 220 DATA "OSMIORNICE"  0E
 250 FOR f=1 TO 5: READ a,b: PRINT AT a,b;"■": BEEP .1,25: PAUSE 10: NEXT f  A6
 260 DATA 15,3,14,7,16,16,14,23,20,29  85
 265 DIM x(3): DIM y(3)  4D
 270 LET y(1)=8: LET x(1)=5: PRINT AT y(1),x(1);"■"  C0
 271 LET y(2)=8: LET x(2)=15: PRINT AT y(2),x(2);"■"  05
 272 LET y(3)=8: LET x(3)=25: PRINT AT y(3),x(3);"■"  19
 273 LET y=1: LET x=15: PRINT AT y,x;"■"  88
 275 LET c=125: LET l=0  F6
 276 LET c=c-1: PRINT AT 0,0;c;" ": IF c=0 THEN GO TO 5000  AD
 290 IF INKEY$="p" AND x<31 THEN GO SUB 3000  2D
 291 IF INKEY$="o" AND x>0 THEN GO SUB 3100  41
 292 IF INKEY$="q" AND y>1 THEN GO SUB 3200  AC
 293 IF INKEY$="a" AND y<21 THEN GO SUB 3300  47
 300 FOR z=1 TO 3: LET r=INT (RND*4+1)  B3
 310 GO SUB 3900+100*r  2F
 320 NEXT z  B5
 400 GO TO 276  4E
3000 IF SCREEN$ (y,x+1)<>" " THEN GO TO 5000  B6
3010 PRINT AT y,x;" ": LET x=x+1: PRINT AT y,x;"■"  19
3020 RETURN  D8
3100 IF SCREEN$ (y,x-1)<>" " THEN GO TO 5000  1D
3110 PRINT AT y,x;" ": LET x=x-1: PRINT AT y,x;"■"  6Q
3120 RETURN  40
3200 IF SCREEN$ (y-1,x)<>" " THEN GO TO 5000  81
3210 PRINT AT y,x;" ": LET y=y-1: PRINT AT y,x;"■"  E6
3220 RETURN  A4
3310 IF SCREEN$ (y+1,x)<>" " AND x=3 AND y+1=15 OR x=7 AND y+1=14 OR x=16 AND y+1=16 OR x=23 AND y+1=14 OR x=29 AND y+1=20 THEN PRINT AT y+1,x;" ": GO SUB 5500: GO TO 3320  46
3315 IF SCREEN$ (y+1,x)<>" " THEN GO TO 5000  F2
3320 PRINT AT y,x;" ": LET y=y+1: PRINT AT y,x;"■"  52
3330 RETURN  13
4000 IF x(z)<30 AND SCREEN$ (y(z)+1,x(z))=" " THEN PRINT AT y(z),x(z);" ": LET x(z)=x(z)+1: IF x(z)>31 THEN LET x(z)=31  CE
4005 PRINT AT y(z),x(z);"■"  2C
4010 RETURN  BD
4100 IF x(z)>0 AND SCREEN$ (y(z),x(z)-1)=" " THEN PRINT AT y(z),x(z);" ": LET x(z)=x(z)-1: PRINT AT y(z),x(z);"■"  64
4110 RETURN  E2
4200 IF y(z)>1 AND SCREEN$ (y(z)-1,x(z))=" " THEN PRINT AT y(z),x(z);" ": LET y(z)=y(z)-1: PRINT AT y(z),x(z);"■"  CD
4210 RETURN  86
4300 IF y(z)<21 AND SCREEN$ (y(z)+1,x(z))=" " THEN PRINT AT y(z),x(z);" ": LET y(z)=y(z)+1: PRINT AT y(z),x(z);"■"  72
4310 RETURN  EA
5000 BEEP 1,-40: PRINT AT y,x;"*"  67
5020 LET l$="KONIEC GRY"  91
5030 PAUSE 30: FOR f=1 TO LEN l$: PRINT AT 10,10;l$(1 TO f): BEEP .1,-28: NEXT f  57
5040 PAUSE 60: RUN  A6
5500 PRINT AT y+1,x;" ": LET l=l+1  27
5505 PRINT AT 0,26+l;"■"  F6
5510 IF l=5 THEN GO TO 6000  83
5520 RETURN  A9
6000 PAPER 7: INK 0: BORDER 7: CLS  01
6010 FOR f=0 TO 11: PRINT AT 10,f;"■": PAUSE 5: PRINT AT 10,f;" ": NEXT f  6B
6020 FOR f=12 TO 16: PRINT AT 10,f;"■": PRINT AT 11,f;"■": PAUSE 5: PRINT AT 10,f;" ": NEXT f  C3
6030 FOR f=17 TO 31: PRINT AT 10,f;"■": PAUSE 5: PRINT AT 10,f;" ": NEXT f  DF
6040 LET l$="BRAWO!"  5E
6050 FOR f=1 TO LEN l$: PRINT AT 11,12;l$(1 TO f): BEEP .1,f+10: NEXT f  E2
6060 PAUSE 30: FOR f=1 TO 30: BEEP .1,RND*20+20: BORDER RND*7: NEXT f  9D
6070 PAUSE 30: INPUT "Jeszcze raz? ";i$  4E
6080 IF i$="n" OR i$="N" THEN RANDOMIZE USR 1333: RANDOMIZE USR 0  5F
6090 RUN  DE
```

```
statek      ... ■
ośmiornica  ... ■
nurek       ... ■
skała       ... ■
```

PROGRAM KOPIUJĄCY DO STACJI DYSKÓW

Istnieje wspaniała zabawka, na opakowaniu której widnieje napis „Personal Computer". Ta zabawka to ZX Spectrum. Projektanci pudełka nie zastanawiali się nad treścią serii reklamowych napisów, dlatego dzisiaj każdy uśmiecha się, widząc bezsprzeczny dowód ignorancji pracowników firmy Sinclair Research Ltd.

Można jednak uczynić sporo, aby możliwości Spectrum zbliżały go do klasy komputerów, do jakiej został zakwalifikowany napisem na pudełku. Pierwszym krokiem jest zakupienie stacji dysków.

U nas, w Polsce najlepiej nabyć stację Timex FDD 3000, w Centralnej Składnicy Harcerskiej. Cena jej jest stosunkowo niska i stacja staje się poważną ofertą dla posiadacza Spectrum, który chciałby poszerzyć swe horyzonty.

Gdy już będziemy szczęśliwymi posiadaczami stacji, możemy czuć się, jak posiadacze „prawie Personal Computera". Czy jednak nie trochę na wyrost?

Cóż z tego, że mamy dobry komputer — przecież każdy wie, że równie ważne co sprzęt jest oprogramowanie. Po paru dniach „zabawy" rzuca się w oczy dotkliwy brak jakichkolwiek programów użytkowych, w szczególności kopiujących.

Napisany przez nas specjalistyczny program kopiujący TeerCopy można uzyskać w redakcji „Bajtka". Nie możemy go opublikować m.in. dlatego, że ma ponad 4 KB. W zamian przedstawiamy prosty program w BASIC-u.

Jest on bardzo prymitywny w porównaniu z TeerCopy, lecz zawiera interesujący trick.

O ile kopiowanie plików kodu maszynowego jest łatwe, to kopiowanie BASIC-a jest niemożliwe, gdyż BASIC wgrywa się zawsze w obszar BASIC-a. Trzeba więc zrobić coś, co pozwoli na wgranie BASIC-a w obszar kodu maszynowego.

Wystarczy zmienić nagłówek na taki, który TOS rozpozna jako plik kodu.

Możliwe to będzie wtedy, gdy plik BASIC-a nie jest zabezpieczony programowo ani mechanicznie (TeerCopy omija tę przeszkodę).

Program zamieszczony obok kopiuje pliki typu 3 (kod) oraz 0 (BASIC). Dokładną analizę pozostawiam Czytelnikom.

Po uważnym wpisaniu listingu należy zgrać go na dysk przez SAVE *"filecopy"LINE 800. Wczytanie do pamięci następuje po wprowadzeniu LOAD *"filecopy".

Gdy posiadamy już stację generator AY, TeerCopy i inne użytki, rozszerzymy pamięć do 80 KB to możemy śmiało stawać w szranki z Atari ST.

Maciej Pietraś

```
   5 REM FileCopy
  10 CLEAR VAL "25499": CAT *: POKE VAL "23617",CODE " INPUT ": INPUT "File:"; LINE f$
  20 POKE VAL "23729",SGN PI: CAT *f$: IF PEEK VAL "23728"<>NOT PI THEN RUN
  40 POKE VAL "23729",NOT PI: OPEN #*SGN PI;f$;i;VAL "7": INPUT *#SGN PI;n$: CLOSE #*
  50 LET typ=CODE n$: LET len=CODE n$(VAL "2")+VAL "256"*CODE n$(VAL "3"): LET st=CODE n$(VAL "4")+VAL "256"*CODE n$(VAL "5")
  60 IF typ=NOT PI THEN GO TO VAL "700"
  70 GO SUB VAL "500": PRINT 'f$;AT VAL "2",VAL "10";"B";AT VAL "2",VAL "15";len;AT VAL "2",VAL "21";st
  80 IF len>VAL "40030" THEN BEEP VAL ".1",CODE "(": PRINT 'TAB VAL "9";"OUT OF MEMORY": GO SUB VAL "610": RUN
  90 LOAD *f$CODE VAL "25500": PRINT #NOT PI;TAB VAL "6";"INSERT TARGET DISK.": GO SUB VAL "610"
 110 INPUT '': SAVE *f$CODE VAL "25500",len: OPEN #*SGN PI;f$;r;VAL "7": PRINT *#SGN PI;n$: CLOSE #*: GO SUB VAL "600": RUN
 500 CLS : PRINT " name:          typ len  start  prog": RETURN
 600 BEEP VAL ".4",VAL "-4": PRINT #NOT PI;TAB VAL "4";"OK, INSERT MASTER DISK."
 610 IF CODE INKEY$<>VAL "13" THEN GO TO VAL "610"
 620 RETURN
 700 LET st=len: LET len=CODE n$(VAL "4")+VAL "256"*CODE n$(VAL "5"): LET pr=CODE n$(VAL "6")+VAL "256"*CODE n$(VAL "7")
 710 GO SUB VAL "500": PRINT 'f$;AT VAL "2",VAL "13";"B";AT VAL "2",VAL "15";len;AT VAL "2",VAL "21";st;AT VAL "2",VAL "27";pr
 715 LET lem=len+VAL "2": RANDOMIZE lem: LET l1=PEEK VAL "23670": LET l2=PEEK VAL "23671": LET m$=CHR$ VAL "3"+CHR$ l1+CHR$ l2
 720 POKE VAL "23729",SGN PI: OPEN #*SGN PI;f$;r;VAL "3": PRINT *#SGN PI;m$: CLOSE #*
 730 POKE VAL "23729",NOT PI: IF PEEK VAL "23728"=CODE "8" THEN PRINT ''"IF YOU WANT COPY A BASIC PROGRAMDISK MUST BE POSSIBLE TO RECORD.": BEEP VAL ".1",CODE "(": GO SUB VAL "610": RUN
 740 LOAD *f$CODE VAL "25500": OPEN #*SGN PI;f$;r;VAL "7": PRINT *#SGN PI;n$: CLOSE #*
 750 PRINT #NOT PI;TAB VAL "6";"INSERT TARGET DISK.": GO SUB VAL "610": INPUT ''
 760 SAVE *f$CODE VAL "25500",lem: OPEN #*SGN PI;f$;r;VAL "7": PRINT *#SGN PI;n$: CLOSE #*: GO SUB VAL "600": RUN
 800 BORDER VAL "7": PAPER VAL "7": INK NOT PI: CLS : GO SUB VAL "600": RUN
```

DWUKANAŁOWY GENERATOR DŹWIĘKÓW

Właściciele Atari i Commodore pękają ze śmiechu, gdy usłyszą żałosne buczenie głośniczka naszego poczciwego Spectrum, mina rzednie im po posłuchaniu „AY-greka". Nie czas tu i miejsce na kłótnie i spory, więc zostawmy atarowców i komodorowców, a sami zajmijmy się Spectrum.

To, że jego „oryginalny" dźwięk pozostawia wiele do życzenia, wiedzą wszyscy. Wystarczy jednak nieco wzmocnić sygnał, z głośniczka (podawany również na wyjście MIC), by otrzymać całkiem niezły dźwięk, rodem jak z pierwszych syntezatorów Yamaha.

Jedyną komendą języka BASIC, służącą do obsługi dźwięku jest BEEP. Za jej pomocą nie można wiele uzyskać i też prawie nikt jej nie używa.

Poniżej chciałbym zaproponować jeden z wielu sposobów uzyskania pseudodwukanałowego generatora dźwięku. Realizuje go procedura w języku maszynowym, umieszczona w pamięci od adresu 65006 i wywoływana przez RANDOMIZE USR 65006. Przedtem jednak należy ustalić dane wejściowe i umieścić je w pamięci następująco:

65000 — częstotliwość I generatora
65001 — częstotliwość II generatora
65002
65003 — czas trwania dźwięku

Podczas działania generatora wstrzymywane jest wykonywanie przerwań maskowalnych, na co trzeba zwrócić uwagę przy ewentualnym wykorzystaniu zmiennej systemowej FRAMES.

Z uwagi na wykorzystanie tylko skoków relatywnych procedura jest relokowalna, lecz zmiana położenia parametrów wejściowych wymaga zmiany treści procedury.

Na koniec podaję szacunkowe wartości częstotliwości generatorów dla dźwięków oktawy rozkreślnej:

C_1 — 98
C# — 92
D — 87
D# — 82
E — 78
F — 73
F# — 69
G — 65
G# — 62
A — 58
A# — 55
H — 52
Cz — 49

Dla uzyskania dźwięków z innych oktaw należy podzielić przez 2 podaną wartość dla oktawy wyższej lub pomnożyć przez 2 dla oktawy niższej

Michał Mosiewicz

```
10 CLEAR 64999
20 FOR N=65006  TO 65062 : READ K
30 POKE N,K: NEXT N: STOP
40 DATA 243,221,33,236,253,42,234
50 DATA 253,221,53,0,221,126,0
60 DATA 32,13,62,24,221,254,58,232
70 DATA 253,221,119,0,175,211,254
80 DATA 221,53,1,221,126,1,32,13
90 DATA 62,24,211,254,58,233,253
100 DATA 221,119,1,175,211,254,43
110 DATA 124,181,32,209,251,201
```

```
 10;dwukanalowy dzwiek
 60           ORG   65000
 70 V1        DEFB  50
 80 V2        DEFB  40
 90 TIME      DEFW  10000
100 COUNT     DEFB  0,0
110;
120 START     DI
130           LD    IX,COUNT
140           LD    HL,(TIME)
150 LOOP      DEC   (IX+0)
160           LD    A,(IX+0)
170           JR    NZ,ET1
180           LD    A,%00011000
190           OUT   (254),A
200           LD    A,(V1)
210           LD    (IX+0),A
220           XOR   A
230           OUT   (254),A
240 ET1       DEC   (IX+1)
250           LD    A,(IX+1)
260           JR    NZ,ET2
270           LD    A,%00011000
280           OUT   (254),A
290           LD    A,(V2)
300           LD    (IX+1),A
310           XOR   A
320           OUT   (254),A
330 ET2       DEC   HL
340           LD    A,H
350           OR    L
360           JR    NZ,LOOP
370           EI
380           RET
```

SIDEKICK
mikrokomputerowy kumpel

W marcowym numerze, odpowiadając na listy Czytelników, Dominik Falkowski obiecał, że zajmiemy się nieco bliżej przerwaniami w Spectrum. Nadszedł czas wywiązania się z obietnicy. Potraktujemy temat dość szeroko — postaramy się opowiedzieć nie tylko o przerwaniach w Spectrum, ale również coś o ich zastosowaniach w innych komputerach.

Zacznijmy od poszukania odpowiedzi na pytanie — po co są przerwania? Jeśli zostały wymyślone i w tej czy innej formie istnieją w każdym komputerze, to znaczy, że są przydatne. Ale do czego? Wyobraźcie sobie następującą sytuację — piszecie program, który ma przeprowadzić bardzo długie obliczenia. Zależy Wam jednak na tym, by móc te obliczenia w dowolnym momencie przerwać. Jeśli używacie któregoś z języków wyższego poziomu — np. BASIC-a, ktoś już kiedyś za Was ten problem rozwiązał — wystarczy nacisnąć klawisz BREAK i sprawa załatwiona. Jeżeli jednak program ma być w kodzie maszynowym, problem staje się skomplikowany. Można sobie próbować z nim poradzić na dwa sposoby — po pierwsze, umieszczając w swoim programie co jakiś czas instrukcję sprawdzającą, czy BREAK nie został naciśnięty — wyobraźnie sobie, o ile dłuższy, wolniejszy i bardziej skomplikowany będzie taki program. Drugą, o wiele prostszą metodą, jest wykorzystanie przerwań — wyobraźcie sobie, że Wasz program jest wykonywany, a w momencie kiedy naciśnięty zostanie klawisz BREAK, procesor (nie program!) zareaguje rzucając aktualnie wykonywane zadanie i zacznie wykonywać specjalny podprogram (znajdujący się w zupełnie innym miejscu pamięci) — tak zwany program obsługi przerwania. Oczywiście procesor musi w jakiś sposób dowiedzieć się, że klawisz BREAK został naciśnięty, ale tym zajęli się już wcześniej jego projektanci — jeśli do jednego z wielu wyprowadzeń mikroprocesora przyłoży się odpowiednie napięcie, zareaguje on wykonując przerwanie.

Oczywiście w analogiczny sposób mikroprocesor może reagować również na inne klawisze i inne sytuacje — np. specjalny klawisz do kopiowania ekranu, albo sygnał od drukarki — „skończyłam drukować poprzednią porcję danych, proszę o jeszcze". Ważne jest, żeby po powrocie z obsługi przerwania procesor znajdował się dokładnie w takim samym stanie, w jakim był przed przerwaniem — w przeciwnym razie zaczną się dziać straszne rzeczy.

To, o czym sobie przed chwilą opowiedzieliśmy, to tylko wierzchołek góry lodowej, gdyż idea przerwań jest bardzo pojemna, i udało się ją wykorzystać do bardzo wielu celów. Jednym z nich jest zapewnienie kompatybilności między różnymi komputerami. Co ma piernik do wiatraka? — otóż ma, i to wcale niemało. Wprawdzie przerwania, którymi się tu posłużono, polegają na czymś trochę innym ale koncepcja jest podobna. Mikroprocesor na określony sygnał wykonuje jakąś czynność — np. drukuje znak na ekranie, po czym wraca do programu tak, jakby nic się nie stało. Tym sygnałem jest tak zwany rozkaz przerwania — ponieważ jest to instrukcja mikroprocesora, którą umieszcza się w programie, ten rodzaj przerwań nosi nazwę przerwań programowych. (Tak naprawdę to przerwania programowe nie różnią się specjalnie od wywoływania podprogramów, tylko że nie trzeba podawać adresu podprogramu — jest on albo ustalony, albo znajduje się w specjalnej tablicy, tak że procesorowi wystarczy znajomość numeru przerwania.) A co ma do tego kompatybilność? Wyobraźcie sobie, że budujecie z kolegą komputery — każdy z Was ma własną koncepcję konstrukcji, ale zależy Wam na tym, żeby programy, które będziecie pisać, mogły być wykonywane na obu maszynach. Wystarczy, że się umówicie — stosujemy ten sam procesor, i wydrukowanie znaku na ekranie to jest wywołanie pierwszego przerwania z kodem znaku w rejestrze A mikroprocesora, a wywołanie drugiego przerwania spowoduje załadowanie przez komputer do tego samego rejestru kodu znaku jaki został wprowadzony z klawiatury. Jeszcze kilka takich ustaleń — i mimo że Wasze komputery będą zupełnie inne będą mogły wykonywać te same programy. Na takich właśnie ustaleniach opierają się systemy CP/M i MS DOS — dokładnie wiadomo, co oznacza które przerwanie i jak z niego korzystać.

Zauważcie, że nasz problem kontrolowania klawisza BREAK można rozwiązać w jeszcze jeden sposób. Jeśli zmusimy mikroprocesor, żeby co jakiś czas, na przykład co 0,02 sekundy wykonywał program przerwania i sprawdzał stan całej klawiatury — nie tylko jednego czy dwóch klawiszy specjalnych, ale wszystkich bez wyjątku, a w pewnych sytuacjach wykonywał pewne dodatkowe operacje, to problem jest rozwiązany, co więcej — mamy kilka nieoczekiwanych profitów. Po pierwsze, nie musimy się martwić, jak zrealizować w naszym komputerze komunikację między klawiaturą a programem — mikroprocesor sprawdzając stan klawiatury wpisze do odpowiedniej zmiennej systemowej kod naciśniętego klawisza, i sygnał „klawisz świeżo naciśnięty", a program skorzysta z tych informacji dopiero wtedy, kiedy będzie mu to potrzebne. Po drugie — procesor może służyć do odmierzania czasu — za każdym razem zwiększając odpowiednią zmienną systemową o jeden — będzie w niej przechowywać informację o tym, ile razy po 0,02 sekundy minęło od początku pracy. Po trzecie — można napisać wiele programów, których inaczej napisać się nie da — ale o tym za chwilę.

Teraz nadszedł czas na zadanie sobie pytania — co to są różne tryby przerwań? Są to — w przypadku procesora Z80 — różne sposoby adresowania podprogramu obsługi przerwania. Brzmi to trochę metnie, toteż zaraz zajmiemy się przykładami, które powinny trochę nam rozjaśnić w głowach. Są trzy różne tryby, ale — z przyczyn konstrukcyjnych — w Spectrum zupełnie nie da się korzystać z pierwszego z nich, toteż pominiemy go w naszych rozważaniach. W przypadku

drugiego trybu (a właściwie pierwszego, ponieważ tryby numerowane są od zera — IM 0, IM 1 i IM 2 — IM 0 to ten nieosiągalny) adres podprogramu przerwania jest ustalony raz na zawsze i wynosi # 0038 — po otrzymaniu sygnału wywołującego przerwanie procesor zaczyna wykonywać podprogram, znajdujący się w tym miejscu pamięci. Tak właśnie jest w Spectrum.

Najciekawszy jest ostatni tryb — IM 2. Tym razem jest zupełnie inaczej — po otrzymaniu sygnału przerwania procesor w pewien sposób konstruuje adres, od którego ma zacząć pracę. Korzysta w tym celu z kilku informacji — jednej zapisanej w rejestrze I, drugiej, którą wczytuje z szyny danych (jest to numer urządzenia, które domaga się obsłużenia) i trzeciej, zapisanej w pamięci. Wyobraźcie sobie na przykład, że w rejestrze I znajduje się liczba 100, a przerwania domaga się drukarka, mająca numer 6. Drukarka musi wysłać do procesora dwa sygnały — jeden o potrzebie wykonania przerwania, drugim jest jej numer, wysłany na szynę danych. Procesor adresu podprogramu przerwania będzie szukać w miejscu pamięci wskazanym przez rejestr I — starszy bajt, i liczba z szyny danych — młodszy bajt. W naszym przypadku będzie to 256∗100+6 czyli 25606 — pod tym adresem w pamięci musi znajdować się starszy bajt adresu podprogramu obsługi przerwania, a młodszy tuż obok — zgodnie z konwencją przyjętą w Z80 pod adresem 25605. Trochę to skomplikowane — adresy adresów, ale tak to jest zrobione, i nie jest to wcale zły pomysł. Zauważcie jeszcze, że numery urządzeń muszą być zawsze parzyste — gdyby jakieś inne urządzenie miało numer 5, to starszy bajt adresu podprogramu obsługi musiałby się znajdować pod adresem 25605 — czyli tam, gdzie jest młodszy bajt adresu podprogramu obsługi drukarki!

Teraz możemy spróbować wykorzystać do czegoś świeżo nabytą wiedzę. Ale do czego? Bez budowania Spectrum od nowa nie uda nam się zmusić drukarki do wysłania na szynę danych swojego numeru, toteż nie możemy wykorzystać trybu IM 2 do obsługiwania różnych urządzeń. Zresztą przerwanie w Spectrum jest generowane 50 razy na sekundę, i nie da się tego zmienić bez następnej ingerencji w konstrukcję. Trzeba więc poszukać innej drogi. Na szczęście taka istnieje — można zamienić istniejącą procedurę obsługi przerwania (tą spod adresu #0038) na własną, umieszczoną pod innym adresem. Wymaga to wprawdzie trochę gimnastyki, ale chyba damy sobie z tym radę.

Wiemy już, że adres naszej procedury musimy umieścić gdzieś w pamięci tak, by procesor mógł go sobie w odpowiednim momencie znaleźć i wykorzystać. Wiemy, jak procesorowi podać starszy bajt adresu naszego adresu — trzeba go wprowadzić do rejestru I. Ale co będzie z młodszym bajtem? Nie wiadomo, co zostanie wczytane z szyny danych — jeżeli dysponujemy gołym Spectrum, bez żadnego interfejsu podłączonego do szyny krawędziowej, będzie to liczba 255 — więc nieparzysta, jeżeli mamy podłączony jakiś interfejs — procesor może za każdym razem wczytywać coś innego. Same kłopoty! (Wbrew temu co podają różne książki, najmłodszy bit liczby wczytanej z szyny danych nie jest ignorowany!) A jeśli by tak adres naszego programu miał identyczny starszy i młodszy bajt — np. #9E9E, i gdyby wartością #9E wypełnić większy obszar pamięci — wszystkie te bajty których adres ma starszy bajt równy zawartości rejestru I? To jest to! Teraz jesteśmy niezależni od tego, co zostanie wczytane przez procesor z szyny danych — wywołany będzie zawsze nasz program. Szkoda tylko, że musimy zmniejszyć obszar pamięci o 256 bajtów — ale i na to znajdzie się rada! W Spectrum jest dość duży obszar pamięci ROM wypełniony liczbą 255 (między adresami 14446 a 15615). Utworzony z dwóch liczb 255 adres nie wygląda może zbyt zachęcająco — w końcu jest to ostatni bajt pamięci (#FFFF), po którym nie ma już nic. Na szczęście nie jest to do końca prawda — następnym bajtem jest bajt #0000 (dlaczego? — powinniście sami wiedzieć, ilu bitowe jest Spectrum). Zawartość adresu #0000 nadaje się do wykorzystania jako argument instrukcji skoku, będzie to wyglądało następująco:

```
#FFFF    #18      JR      #FFF4
#0000    #F3
```

Nasz program obsługi przerwania będzie się więc zaczynał pod adresem #FFF4.

A co może robić nasz program? Na przykład przewijać część ekranu, nie zwracając najmniejszej uwagi na to, co robi Spectrum, albo grać jakąś muzyczkę — bardzo często tak jest w grach. My zrobimy coś innego — pierwszego na świecie SideKicka na Spectrum. Co to jest SideKick? Po amerykańsku to kumpel, a po komputerowemu — specjalny program ułatwiający życie. Został on napisany przez bardzo znaną firmę BORLAND (tą od Turbo Pascala i nie tylko) dla komputerów kompatybilnych z IBM PC. Zasada jest bardzo prosta — SideKick cały czas czuwa w pamięci, a po naciśnięciu odpowiedniej kombinacji klawiszy pojawia się na ekranie — niezależnie od tego, co akurat robiliśmy. Przy jego pomocy można coś policzyć (również w systemie heksadecymalnym i binarnym) — zawiera kalkulator, można sprawdzić kod ASCII dowolnego znaku, można napisać list — zawiera edytor tekstów, można również zrobić kilka innych rzeczy. A wszystko to jest cały czas pod ręką, gotowe do akcji. Ci z Was, którzy piszą dużo programów, wiedzą już, dlaczego SideKick jest taki wygodny — zastępuje kartkę i ołówek, których trzeba przy pracy co chwilę używać — zapisać jakiś adres, coś policzyć — dodać, odjąć, podzielić..., i nie trzeba się miotać po pokoju, żeby znaleźć instrukcję z kodami ASCII... Same plusy — toteż SideKicka używa bardzo wiele osób. Nasz SideKick z konieczności będzie bardzo skromny — inaczej wydruk programu musiałby zajmować cały numer Bajtka. Jedyne co przy jego pomocy można zrobić, to obejrzeć tabelkę kodów ASCII wszystkich znaków. Żeby ją zobaczyć wystarczy nacisnąć na dwie sekundy klawisz shift — i już, tabelka jest widoczna — dopóki shift jest naciśnięty. W momencie, kiedy się go puści, Spectrum wraca do dokładnie takiego samego stanu, w jakim było przed wywołaniem SideKicka.

Program jest opisany komentarzami, ale przyda się kilka słów o tym jak działa. Po wywołaniu program sprawdza, czy klawisz shift jest naciśnięty — jeśli nie, to do licznika wprowadzana jest liczba 101, jeżeli tak — zawartość licznika jest zmniejszana o jeden, i — jeżeli okaże się że licznik został wyzerowany — uruchamiany jest sam SideKick. Licznikiem jest bajt o adresie 23728 (IY+118, bo IY zawsze zawiera 23610) — nieużywany, a leżący w obszarze zmiennych systemowych. Potrzebne jest jeszcze kilka informacji — program ładujący (trzeba się nim posłużyć) przygotowuje (w liniach 220–280) ekran który będzie wywoływany z pamięci — nie wolno mu w tym przeszkadzać, gdyż instrukcja POKE 23659,0 uniemożliwia prawidłowe zatrzymanie programu (Spectrum się zawiesi). (Po linii 300 program może się już zatrzymać bez żadnych obaw). Wykonanie instrukcji NEW nie kasuje naszego SideKicka, chociaż go dezaktywuje — Spectrum jest przełączane w tryb IM 1. Żeby program uaktywnić ponownie należy użyć instrukcji RANDOMIZE USR 65453. Dodatkowego wytłumaczenia może wymagać fakt wychodzenia z naszego podprogramu instrukcją, JP#0038 — ale wystarczy się zastanowić nad tym, że gdzieś trzeba sprawdzić stan klawiatury. Oprócz tego procedura zawarta w ROM-ie zrobi dla nas jeszcze jedną ważną rzecz. Po wykonaniu przerwania procesor nie może przyjąć następnego przerwania dopóki nie natrafi na rozkaz, który mu na to pozwoli — inaczej mogłoby się tak zdarzyć, że przerwanie zostałoby przerwane przerwaniem, które zostałoby przerwane przerwaniem, które..., a potem trzeba by zaczynać wszystko od początku. Przedostatnim rozkazem procedury z ROM-u jest właśnie rozkaz pozwalający na przyjmowanie przerwań (EI). To wszystko.

Marcin Borkowski

```
 10           ORG     65435
 20 INSTAL    LD      HL, 15880       ;Procedura
 30 BUFOR     EQU     INSTAL-6912;instalująca.
 40 UDG       EQU     BUFOR-168
 50           LD      DE, UDG         ;Przygotowanie
 60           LD      (23675), DE     ;obszaru pod
 70           LD      BC, 168         ;UDG.
 80           LDIR
 90           CALL    TRANSP
100 INST2     LD      A, #39          ;Rejestru I nie można
110           LD      I, A            ;załadować bezpośrednio.
120           IM      2               ;Przełączenie w tryb IM 2
130           RET
140 SKICK     EX      AF, AF'         ;Przechowaj na stosie
150           PUSH    AF              ;zawartość wszystkich
160           PUSH    BC              ;używanych rejestrów.
170           PUSH    DE
180           PUSH    HL
190           CALL    TRANSP          ;Pokaż tabelkę.
200 SKP       LD      A, 127          ;Sprawdź stan klawisza
210           IN      A, (254)        ;shift, jeśli
220           BIT     1, A            ;naciśnięty sprawdź
230           JR      Z, SKP          ;ponownie.
240           CALL    TRANSP          ;Schowaj tabelkę.
250           POP     HL              ;Odzyskaj zawartości
260           POP     DE              ;rejestrów.
270           POP     BC
280           POP     AF
290           EX      AF, AF'
300           RET
310 TRANSP    LD      HL, #4000       ;Podprogram TRANSP
320           LD      DE, BUFOR       ;zamienia miejscami
330           LD      BC, 6912        ;zawartość dwóch
340 TR1       LD      A, (DE)         ;obszarów pamięci:
350           EX      AF, AF'         ;ekran i atrybuty oraz
360           LD      A, (HL)         ;bufor.
370           LD      (DE), A         ;Są to tabelka, i
380           EX      AF, AF'         ;zapamiętany obraz
390           LD      (HL), A         ;przy którym uzytkownik
400           INC     HL              ;wywołał SideKicka.
410           INC     DE              ;HL - obraz,
420           DEC     BC              ;DE - bufor,
430           LD      A, B            ;BC - licznik bajtów.
440           OR      C
450           JR      NZ, TR1
460           RET
470 CONT      JR      Z, TAK          ;Jeśli shift nie nac.
480           LD      (IY+118), 101;odtwórz licznik.
490 TAK       DEC     (IY+118)        ;Zmniejsz licznik, jeśli
500           CALL    Z, SKICK        ;zero - wywołaj SKicka.
510           POP     AF              ;Odzyskaj AF.
520           JP      #0038           ;Wywołaj ROM.
530 INTRPT    PUSH    AF              ;#FFF4 - przechowaj AF
540           LD      A, 127          ;sprawdź co z
550           IN      A, (254)        ;klawiszem shift.
560           BIT     1, A
570           JR      CONT            ;Brak miejsca - skocz
580           NOP                     ;kawałek wcześniej.
590           NOP
600 FFFF      JR      INTRPT          ;Początek!
```

```
 10 CLEAR 58354: LET r=65435
 20 FOR i=r TO r+90 STEP 10
 30 LET s=0
 40 FOR j=0 TO 9
 50 READ a: POKE i+j, a
 60 LET s=s+a
 70 NEXT j
 80 READ a: IF s<>a THEN PRINT "Błąd w lini ";100+i-r
 90 NEXT i
100 DATA 33, 8, 62, 17, 243, 227, 237, 83, 123, 92, 1125
110 DATA 1, 168, 0, 237, 176, 205, 205, 255, 62, 57, 1366
120 DATA 237, 71, 237, 94, 201, 8, 245, 197, 213, 229, 1732
130 DATA 205, 205, 255, 62, 127, 219, 254, 203, 79, 40, 1649
140 DATA 248, 205, 205, 255, 225, 209, 193, 241, 8, 201, 1990
150 DATA 33, 0, 64, 17, 155, 228, 1, 0, 27, 26, 551
160 DATA 8, 126, 18, 8, 119, 35, 19, 11, 120, 177, 641
170 DATA 32, 243, 201, 40, 4, 253, 54, 118, 101, 253, 1299
180 DATA 53, 118, 204, 180, 255, 241, 195, 56, 0, 245, 1547
190 DATA 62, 127, 219, 254, 203, 79, 24, 231, 0, 0, 1199
200 POKE 65535, 24: POKE 23675, 8
210 POKE 23676, 62: POKE 23659, 0
220 PRINT "Spectrum SideKick  © Bajtek 1989"
230 FOR i=0 TO 5
240 FOR j=0 TO 21
250 LET c=32+22*i+j
260 PRINT AT j+1, 5*i+(c>119)+(c>141);CHR$ c;"-";c
270 NEXT j
280 NEXT I
290 RANDOMIZE USR 65435
300 POKE 23659, 2
310 PAPER 7: INK 0: CLS
```

JĘZYK

C DLA NAJMŁODSZYCH

czyli czwarty wieczór z czarnoksiężnikiem

Nasz dobry znajomy, majster — czarnoksiężnik, postanowił dokładniej zapoznać się z wcześniej poznanymi elementami języka „C".

— Przede wszystkim, powinienem znać wszystkie możliwe sposoby zapętlenia programu — pomyślał. Otworzył czarnoksięską księgę i zaczął czytać:

„Kolejnym sposobem zapętlenia programu jest wykonanie instrukcji w pętli „wykonuj". Ma ona postać:

```
wykonuj
        instrukcja;
dopoki warunek;
```

Realizacja tej pętli powoduje wykonanie „instrukcji" i ewentualne powtarzanie jej dopóty, dopóki „warunek" jest prawdziwy (różny od zera).

Czarnoksiężnik zamknął księgę i zamyślił się. Rzeczywiście, są sytuacje, w których pewną czynność należy wykonać co najmniej jeden raz. Klasycznym przykładem może być zmiana liczby na ciąg reprezentujących ją cyfr. Niezależnie od wartości tej liczby, trzeba napisać jedną cyfrę i ewentualnie następne.

Majster odłożył księgę i usiadł do komputera. Musiał przecież uzupełnić zbiór magicznych zaklęć. Dopisał do istniejącego już zbioru ‹predef.h› następne zaklęcia:

```
#define wykonuj do
#define dopoki while
```

Po zakończeniu modyfikacji zbioru ‹predef.h› nasz przyjaciel zamyślił się.

— Wszystkie trzy rodzaje pętli mają wspólną cechę. Każda z nich zawiera element „warunek", którego wartość decyduje o tym, czy wykonywany jest kolejny cykl pętli, czy następna za pętlą, instrukcja. Znam jeden sposób konstruowania „warunku". Zapewne istnieją też inne.

Wziął księgę do rąk i kontynuował czytanie:

„Element ‚warunek' może mieć jedną z dwu postaci:

1° — może być dowolnym wyrażeniem logicznym, które porównuje dwa wyrażenia arytmetyczne. W szczególności może to być porównanie zmiennej z jakąś liczbą. Wyrażenie logiczne informuje nas o tym, w jakiej relacji pozostają porównywane wielkości.

2° — może być dowolnym wyrażeniem arytmetycznym, zawierającym dowolne funkcje (te które znamy i te które poznamy później). W takim przypadku stosuje się zasadę, że warunek jest fałszywy, jeśli wartość wyrażenia równa jest zero. W każdym innym przypadku „warunek" jest prawdziwy".

— W obu przypadkach powinienem wiedzieć, jakie operatory logiczne i arytmetyczne mogą tworzyć „warunek" — pomyślał i z dużym zainteresowaniem kontynuował czytanie:

„W języku „C" istnieją operatory jednoargumentowe, dwuargumentowe i trzyargumentowe.

Operatory jednoargumentowe:

*wyraż	— jest to operator, odpowiednik adresowania pośredniego. Będzie on szczegółowo omówiony podczas następnych spotkań.
&wyraż	— wskaźnik do obiektu. Operator spokrewniony z poprzednim, także będzie omówiony później.
—wyraż	— zmienia wartość wyrażenia z dodatniej na ujemną i odwrotnie, z ujemnej na dodatnią (arytmetyczny minus).
!wyraż	— dla wyraż = 0 daje jeden, zaś dla wyraż ≠ 0 daje zero.
~wyraż	— logiczna negacja. W binarnym zapisie „wyraż" wszystkie zera zmieniane są na jedynki a jedynki na zera.
++zmienna	— preinkrementacja, zwiększa wartość zmiennej o jeden, po czym udostępnia ją do obliczeń.
--zmienna	— predekrementacja, zmniejsza wartość zmiennej o jeden, po czym udostępnia ją do obliczeń.
zmienna++	— postinkrementacja, udostępnia wartość zmiennej do obliczeń, po czym zwiększa jej wartość o jeden.
zmienna--	— postdekrementacja, udostępnia wartość zmiennej do obliczeń, po czym zmniejsza ją o jeden.
(nazwa_typu)wyrażenie	— typ wartości wyrażenia będzie przekształcony na typ, którego nazwę ujęto w nawiasy. Operator ten nosi nazwę rzutu.

sizeof wyrażenie — podaje wielkość pamięci zajętej przez wyrażenie

sizeof(nazwa_typu) — podaje rozmiar obiektów wskazanego typu.

Operatory dwuargumentowe:

w1*w2	mnożenie w1 i w2. Nie mylić z operatorem wskaźnikowym z poprzedniej grupy.
w1/w2	dzielenie. Jeśli obydwa wyrażenia są typu „całkowite" to wynik jest zaokrąglony.
w1%w2	reszta z dzielenia w1 przez w2. Nie mylić ze specyfikacją wyrażenia w instrukcji „pisz".
w1+w2	dodawanie w1 i w2
w1−w2	odejmowanie w2 od w1
w1≪w2	przesunięcie w lewo w1 jako wzorca binarnego o ilość bitów bądących wartością w2.
w1≫w2	przesunięcie w prawo
w1<w2	operator porównania w1 mniejsze od w2
w1>w2	w1 większe od w2
w1<=w2	w1 mniejsze lub równe w2
w1>=w2	w1 większe lub równe w2
w1==w2	w1 równe w2
w1!=w2	w1 różne od w2
w1&w2	bitowy iloczyn. Każdy bit wyniku równy jest jeden, gdy odpowiednie bity w w1 i w2 są równe jeden, jeśli zaś nie to bit ten równy jest zero.
w1∧w2	różnica symetryczna. W w1 zmieniane są z zera na jedynki i z jedynki na zera, te bity, które mają wartość jeden w w2.
w1|w2	suma logiczna. Wynik ma jedynki na tych bitach, które są równe jeden w w1 lub w w2
warunek1&&warunek2	iloczyn warunków. Wynik jest prawdziwy gdy obydwa warunki są prawdziwe.
warunek1 || warunek2	suma warunków. Wynik jest prawdziwy jeśli jeden z warunków jest prawdziwy.
zmienna = wyrażenie	instrukcja podstawienia. Zmienna przyjmuje wartość wyrażenia.

Instrukcja podstawienia może być poprzedzona operatorem. Wtedy nową wartością zmiennej jest stara wartość potraktowana wyrażeniem zgodnie z operatorem. W szczególności:

zmienna -= wyrażenie	nową wartością zmiennej jest jej stara wartość minus wyrażenie.
zmienna*=wyrażenie	nową wartością zmiennej jest jej stara wartość pomnożona przez wyrażenie.
zmienna /= wyrażenie	nową wartością zmiennej jest jej stara wartość podzielona przez wyrażenie.
zmienna %= wyrażenie	nową wartością zmiennej jest reszta z podzielenia starej wartości przez wyrażenie.
zmienna ≫= wyrażenie	nową wartością zmiennej jest jej stara wartość przesunięta w prawo ilość bitów równą wartość wyrażenia.
zmienna ≪= wyrażenie	nową wartością zmiennej jest jej stara wartość przesunięta w lewo o ilość bitów równą wartości wyrażenia.
zmienna &= wyrażenie	nową wartością zmiennej jest jej stara wartość pomnożona logicznie przez wartość wyrażenia.
zmienna ∧= wyrażenie	nową wartością zmiennej jest jej różnica symetryczna starej wartości i wyrażenia.
zmienna |= wyrażenie	nową wartością jest suma logiczna starej wartości i wyrażenia.

Dzięki ostatniej konstrukcji można instrukcję:

a = a + 3

zapisać w bardziej zwięzłej formie:

a += 3

Program napisany z wykorzystaniem takich konstrukcji będzie zajmował mniej miejsca w pamięci operacyjnej i będzie wykonywał się szybciej.

Operator trzyargumentowy:

w1 ? w2 : w3 Jeśli w1 jest różne od zera to wynikiem jest wartość w2, w przeciwnym wypadku wynikiem jest wartość w3.

— Teraz chętnie napisałbym program, który zamienia liczbę na ciąg reprezentujących ją znaków i ciąg cyfr zamienia na liczbę. Muszę jednak poznać jeszcze instrukcje wykonywane warunkowo — pomyślał.

W tej chwili w telewizji rozpoczął się kolejny odcinek serialu „Smurfy" z jego ulubionym bohaterem — Gargamelem. Koniec na dziś.

Mieczysław Płacheta

WSZYSTKO DLA WSZYSTKICH

Columbia na ploterze Sony

Większość programów graficznych oferowanych na komputer IBM PC pozwala na współpracę z różnymi ploterami. Jeśli urządzenie to nie jest podłączone do komputera, to możliwe jest wykonanie rysunku na plik dyskowy. Zbiór ten zawierający najczęściej polecenia języka HPGL, może być przesłany później na ploter.

Jeśli dysponujemy ploterem PRN C41 f-my SONY, to współpraca ze wspomnianymi wcześniej programami, wymaga zastosowania translatora komend HPGL na polecenia rozpoznawane przez to urządzenie. Opisywany w majowym numerze Bajtka program MSONY został adaptowany również na komputer IBM PC. Obecne jego wersje umożliwiają działanie z programami AutoCAD, Grapher i SlideWrite.

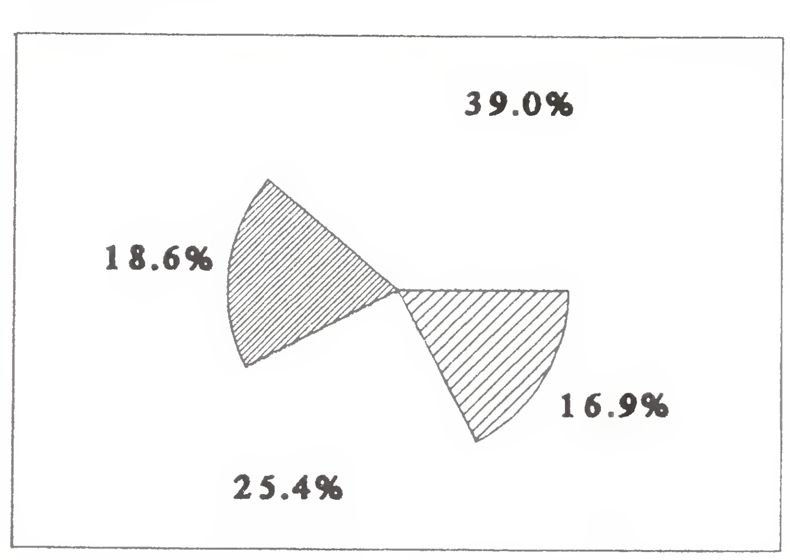
Na rys. 1, 2, 3 przedstawiono obrazki otrzymane na ploterze SONY przy pomocy wymienionych wyżej programów. Ich jakość jest porównywalna z rysunkami uzyskanymi na drukarce, ale czas ich wykonania jest 2-3 razy krótszy. Oczywiście ploter firmy SONY, nie zastąpi 10-20 krotnie droższych urządzeń profesjonalnych, ale w zastosowaniach domowych i półprofesjonalnych jest to bardzo cenny sprzęt.

Niestety z przykrością musimy stwierdzić, że obecnie nie jest możliwe nabycie w CSH plotera SONY, ponieważ zostały one całkowicie wykupione w bardzo krótkim czasie, po ukazaniu się testu w Bajtku. Możemy pocieszyć naszych Czytelników, że podobne urządzenia są wytwarzane w dużych ilościach przez polską firmę Mera-Poltik, ale cała produkcja skierowana jest na Wschód. W Polsce sprzęt ten oferowany jest w ilościach śladowych i do dziś, mimo ośmiomiesięcznych starań, nie udało nam się namówić producenta na udostępnienie redakcji egzemplarza testowego. Należy żałować, że wspomniane plotery, o bardzo dużych walorach dydaktycznych, nie są masowo oferowane polskiemu szkolnictwu.

Jonasz Mayer

Chcę podziękować Panu Markowi Orlikowi za przygotowanie prezentowanych w artykule rysunków i współpracę przy adaptacji programu MSONY na komputer IBM PC.

Rys. 1. Obrazek Columbii uzyskany przy pomocy programu AUTOCAD (ver. 2.62) firmy Autodesk.



Rys. 2. Wykres kołowy uzyskany przy pomocy programu SlideWrite Plus (ver. 2.0) firmy Advanced Graphics Software Inc.

40-ścieżkowa stacja 5 1/4 cala do Joyce'a



Amstrad PCW 8256 wyposażony jest standardowo w jednostronny 3" napęd dysków elastycznych o pojemności 2 ×173 KB.

Producent komputera przewidział też możliwość dodania drugiej dwustronnej stacji 80-ścieżkowej o pojemności 706 KB. Stacja ta, również 3 calowa, instalowana jest po wyjęciu zaślepki znajdującej się poniżej pierwszego napędu. Niestety rozmiar dyskietki stosowany przez firmę Amstrad jest dość nietypowy, a same dyskietki znacznie droższe od bardziej rozpowszechnionego formatu 5 1/4 cala.

Optymalnym rozwiązaniem tego problemu jest dołączenie do Joyce'a zewnętrznego napędu dyskowego przystosowanego do pracy z dyskietkami 5 1/4 cala. Zastosowanie 80-ścieżkowej stacji nie wymaga żadnej zmiany w oprogramowaniu systemowym i zostało przetestowane przeze mnie już dość dawno. Natomiast przez długi czas nie mogłem sobie poradzić z napędem 40-ścieżkowym. Oprogramowanie pisane specjalnie do formatowania dyskietki według zadanego formatu nie zdało egzaminu. Obawiałem się również, że niemożliwe będzie korzystanie z takiej stacji przy pracy z Locoscriptem.

Genialne w swej idei i prostocie rozwiązanie tych problemów, polegające na modyfikacji zbioru systemowego J14 CPM3.EMS i programu DISCKIT.COM, zostało mi przedstawione przez pana Krzysztofa Cieślikowskiego, którego firma oferuje opisaną poniżej 40-ścieżkową stację 5 i 1/4 cala do Joyce'a, łącznie z niezbędnym oprogramowaniem.

SPRZĘT

Napęd dyskowy razem z zasilaczem został umieszczony we wspólnej obudowie metalowej o rozmiarach 150×300×70 mm. Napięcia 5 i 12 V, otrzymane z transformatora sieciowego, są prostowane przez dwa mostki Graetza i stabilizowane układami 7805 i 7812. Kabel łączący z komputerem zakończony z jednej strony złączem krawędziowym, a od strony Amstrada wtykiem męskim typu CANON DB 25, został wykonany zgodnie ze standardem dodatkowej stacji do Bondwella, oferowanej przez CSH. W samym komputerze konieczne było wykonanie połączenia między płaskim wtykiem igłowym, przewidzianym dla dodatkowego, wewnętrznego napędu, a gniazdkiem typu CANON DB 25, mocowanym w dziurze wyciętej na tylnej ściance komputera, obok wyjścia na drukarkę. Dzięki takiemu rozwiązaniu możliwe jest łatwe odłączenie stacji od Amstrada np. przy transporcie. Kieszeń napędu znajduje się na wysokości 6 cm od podstawy i pozwala na wygodną wymianę dyskietek. Przełącznik sieciowy został umieszczony na tylnej ściance obudowy.

Gotowość do pracy sygnalizowana jest żółtą diodą świecącą, która znajduje się na przednim panelu urządzenia. Całość waży mniej niż 3 kg.

OPROGRAMOWANIE

Stacja dostarczana jest razem z przerobionym systemem operacyjnym (zbiór typu EMS) i zmodyfikowanym programem DISCKIT.COM. Oprogramowanie to pozwala na dwustronne sformatowanie dyskietki w 40-ścieżkowym napędzie 5 i 1/4 cala. Tak przygotowaną dyskietką można posługiwać się zarówno pod niezmienionym systemem CP/M Plus, jak i pod Locoscriptem (rys. 1). Komputer automatycznie rozpoznaje nowy format, którego parametry, uzyskane przy pomocy nierezydentnej komendy SHOW B:[drive], pokazano na rysunku 2.

Całkowita pojemność dyskietki 5 1/4 cala wynosi 360 KB, z czego 6 KB zarezerwowane jest na ścieżkę systemową, a 8 KB na katalog (ang. directory). Na zbiory użytkownika, których może być 256, pozostaje 346 KB. Minimalna pamięć zajęta przez plik, tak zwana jednostka alokacji pamięci dyskowej, wynosi 2KB. Na rysunku 3 przedstawiono skorowidz zapełnionej całkowicie dyskietki.

Dla porównania, format 3 calowy stacji jednostronnej pozwala zapamiętać 64 zbiory o łącznej pojemności 173 KB. Jednostka alokacji pamięci wynosi w tym przypadku 1KB. Ewentualne zastosowanie napędu 80-ścieżkowego w miejsce stacji B: daje 706KB na pliki użytkownika, przy niezmienionych innych parametrach w stosunku do dysku 40-ścieżkowego i powinno być rozważone, jeśli zależy nam na dużej pamięci masowej, np. bazy danych. Należy jednak pamiętać, że konieczne w takim przypadku dyskietki HD (High Density), są dwu-, trzykrotnie droższe od typowych dyskietek DD (Double Density).

Współpraca nowej stacji z całym posiadanym przeze mnie oprogramowaniem przebiegała bez zarzutu. Wszystkie programy funkcjonowały poprawnie, a subiektywne wrażenie wolniejszego działania pięciocalowego napędu, spowodowane dwukrotnie większą pojemnością dyskietki, nie zostało potwierdzone przez testy czasowe. Formatowanie jednej strony 3" calowej dyskietki trwa 34 s, natomiast taka sama operacja dla dwóch stron stacji B zajmuje 65 s. Podobnie, kopiowanie plików z ramdysku na obie stacje pozwoliło stwierdzić, że stacja 5 — calowa zapisuje zbiory z szybkością rzędu 2 KB/s i jest szybsza o 15% od standardowego napędu Amstrada.

WIĘKSZY RAMDYSK, CZY DRUGA STACJA?

Typowa pamięć RAM komputera PCW 8256 wynosi 256 KB i może być zwiększona o następne 256KB. W wyniku tej operacji następuje przyrost pojemności ramdysku ze 112 do 368 KB. Tak duża pamięć RAM, pozwala na znaczne przyspieszenie działania wielu programów użytkowych i ułatwia istotnie ich wykorzystanie. Rozszerzenie pamięci RAM kosztuje około 80 $ (cena ośmiu układów 256 Kbit).

Z kolei najważniejszą zaletą drugiej, pięciocalowej stacji dysków, oferowanej za 650 000 zł, jest możliwość przechowywania informacji na znacznie tańszych dyskietkach. Dysproporcje cenowe i mała podaż dyskietek 3" calowych powodują, że przy korzystaniu z 40 sztuk, dodatkowa stacja amortyzuje się całkowicie. Pojemność napędu 5 i 1/4 cala porównywalna jest z wielkością rozszerzonego ramdysku. Następną zaletą tego rozwiązania jest potencjalna możliwość wymiany zbiorów między PCW a innymi komputerami wyposażonymi w stacje 5-calowe (IBM PC, ATARI 65/XE, Commodore 64, D128, itp.).

Biorąc pod uwagę powyższe fakty należy stwierdzić, że jeśli nie stać nas na obie rzeczy, to zakup drugiego napędu jest bardziej opłacalny i należy polecić go każdemu użytkownikowi Joyce'a. Jedyną wadą opisywanego rozwiązania, mam nadzieję, że przejściową, jest brak oprogramowania do wymiany plików z innymi dostępnymi na naszym rynku komputerami.

ZALETY:

+ możliwość korzystania z tanich dyskietek
+ dwukrotnie większa pojemność w stosunku do standardowego napędu
+ wspólny program formatujący dla obu stacji (zmodyfikowany DISCKIT)
+ zgodność okablowania ze standardem komputera Bondwell 8.

WADY:

– brak oprogramowania do wymiany informacji z innymi komputerami.

Jonasz Mayer

```
                    Disc management.                    Printer idle.  Using none.
Printer: Online   Top of form              Idle                High quality   Single sheet
f1=Options   f2=Paper   f3=Actions   f5=Document/Reprint   f7=Reset   f8=On/Off Line   EXIT
Drive A:                         |Drive B:                          |Drive M:
130k used   43k free   26 files  | 0k used  346k free     0 files   | 4k used  352k free    2 files
LETTERS   97k     group 4    0k  | group 0    0k   group 4     0k   |LETTERS     2k   group 4    0k
SAMPLES   12k     group 5    0k  | group 1    0k   group 5     0k   |SAMPLES     0k   group 5    0k
CONT       1k     group 6    0k  | group 2    0k   group 6     0k   |CONT        2k   group 6    0k
TEMPLATE  20k     group 7    0k  | group 3    0k   group 7     0k   |TEMPLATE    0k   group 7    0k
```

```
A>show b:[drive]

    B: Drive Characteristics
2,832: 128 Byte Record Capacity
  354: Kilobyte Drive  Capacity
  256: 32 Byte  Directory Entries
  256: Checked  Directory Entries
  256: Records / Directory Entry
   16: Records / Block
   36: Sectors / Track
    1: Reserved  Tracks
  512: Bytes / Physical Record
```

Rys. 2. Parametry formatu dla stacji 5 i 1/4 cala

```
Directory For Drive B:  User  0

    Name      Bytes   Recs   Attributes      Name      Bytes   Recs   Attributes
------------ ------ ------ ------------  ------------ ------ ------ ------------
BASIC    COM    28k    224 Dir RW        BIN      KEY     2k      2 Dir RW
BINEDI   COM     6k     48 Dir RW        BT               2k     16 Dir RW
COMHEX   COM    10k     73 Dir RW        COMP     COM     2k     10 Dir RW
DDTZ     COM    10k     77 Dir RW        DEVICE   COM     8k     58 Dir RW
DIR      COM    16k    114 Dir RW        DISCKIT  COM     8k     56 Dir RW
ERASE    COM     4k     29 Dir RW        FF       COM    14k    110 Dir RW
GB               2k     16 Dir RW        GENCOM   COM    16k    116 Dir RW
GET      COM     8k     51 Dir RW        HEXCOM   COM     2k      9 Dir RW
INITDIR  COM    32k    250 Dir RW        KNIFE+   COM     6k     48 Dir RW
LINK     COM    16k    123 Dir RW        M80      COM    18k    144 Dir RW
MAIL232  COM     6k     33 Dir RW        NORM     KEY     2k     11 Dir RW
PIP      COM    10k     68 Dir RW        PL               2k     16 Dir RW
PROFILE  SUB     2k      3 Dir RW        PSS      COM    14k    105 Dir RW
PUT      COM     8k     55 Dir RW        R        SUB     2k      1 Dir RW
SAVE     COM     2k     14 Dir RW        SET      COM    12k     81 Dir RW
SETDEF   COM     4k     32 Dir RW        SETSIO   COM     2k     16 Dir RW
SF               2k     16 Dir RW        SHOW     COM    10k     66 Dir RW
SK       COM     2k     16 Dir RW        SP       COM    10k     71 Dir RW
SUBMIT   COM     6k     42 Dir RW        SWAP     COM     2k      7 Dir RW
TIME     COM     2k     10 Dir RW        TURBO    COM    32k    241 Dir RW
TURBO    KEY     2k      9 Dir RW        TURBO    MSG     2k     12 Dir RW

Total Bytes     =    346k  Total Records =    2499  Files Found =   42
Total 1k Blocks =    331   Used/Max Dir Entries For Drive B:   43/ 256
```

Rys. 3. Skorowidz zapełnionej dyskietki 5 i 1/4 cala.


TOTAL ECLIPSE
KHEPRESH
RAMESSES D
PHARAOHS A
SHABAKA
PHARAOHS E
SAHARA A
ILLUSION F
ILLUSION
SAHARA A
RAMESSES B
SAHARA B
HORAKHTY A
START
HORAKHTY D
PHARAOHS H
PHARAOHS F
NEPHTHYS D
PHARAOHS F
NEPHTHYS F
NEPHTHYS G
SAHARA B
RAMESSES
KHERRESH D
NEPHTHYS A
NEPHTHYS B

Mało kto wie, dlaczego żyje. Ale jeden człowiek wie, dlaczego żyje jeszcze ktokolwiek. Tym człowiekiem jest Egon Luisse, dziewięćdziesięcioletni staruszek. Żyje samotnie w małej chatce na Krecie. Obok chatki, w komórce, murszeje i rozpada się coraz bardziej stary samolot. 49 lat temu obaj byli głównymi postaciami dramatu, który mógł skończyć się prosto: nikt by o niczym nie pamiętał, gdyż pamięć 0 bajtów może pomieścić tyleż informacji.

Egon Luisse urodził się 1 kwietnia 1900 roku w Aachen. Taki dzień i taki rok. Być może dlatego jedynym zamiłowaniem Egona była astrologiczna odmiana astronomii. Dzisiaj zarabiałby z łatwością pisząc horoskopy do gazet, wtedy wyklęto go z rodziny.

Miotany przez los po całym świecie, bez przerwy obserwując niebo, znalazł się w Malborku. I tam właśnie dokonał straszliwego odkrycia. Był rok 1930 i to było najgorsze. Zbliżało się bowiem zaćmienie Słońca. I miało być to całkowite zaćmienie. W momencie, gdy tarcza Księżyca przykryje tarczę Słońca i ani jeden promień nie dotrze do Ziemi, miał nastąpić wybuch piramidy Cheopsa. I nie samej piramidy, bo wraz z nią całej kuli ziemskiej. Tak działa klątwa boga Ra.

Jest i pozostanie tajemnicą Egona, w jaki sposób odkrył on to, co odkrył. Wiedział to tylko on i do dzisiaj nikt więcej.

Czym prędzej ukradł więc samolot z lotniska wojskowego w Gdańsku i pognał do Egiptu. Nikt nie strącił go po drodze i możemy teraz tylko dziękować ówczesnym radarom.

Równie mądry, jak głupi stanął Egon przed piramidą. Było południe; Słońce świeciło z góry i piramida rzucała cień zerowy. Dlaczego Egon strzelił ze starego pistoletu właśnie w tę najciemniejszą płytę, pozostanie zawsze tajemnicą. Dość, że płyta otworzyła się, ukazując wejście.

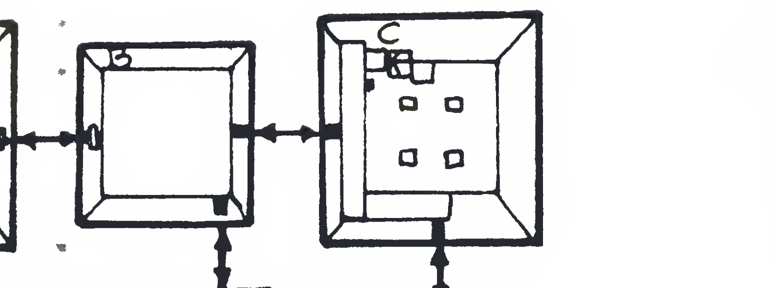
Co dalej robił Egon, nie wiadomo. Nie opowie nikomu o tym, co widział wewnątrz, gdyż od wyjścia z piramidy nic nie mówi. I umrze w milczeniu, pozostawiając po sobie ten kawałek mapy i niesłusznie trwającą ludzkość.

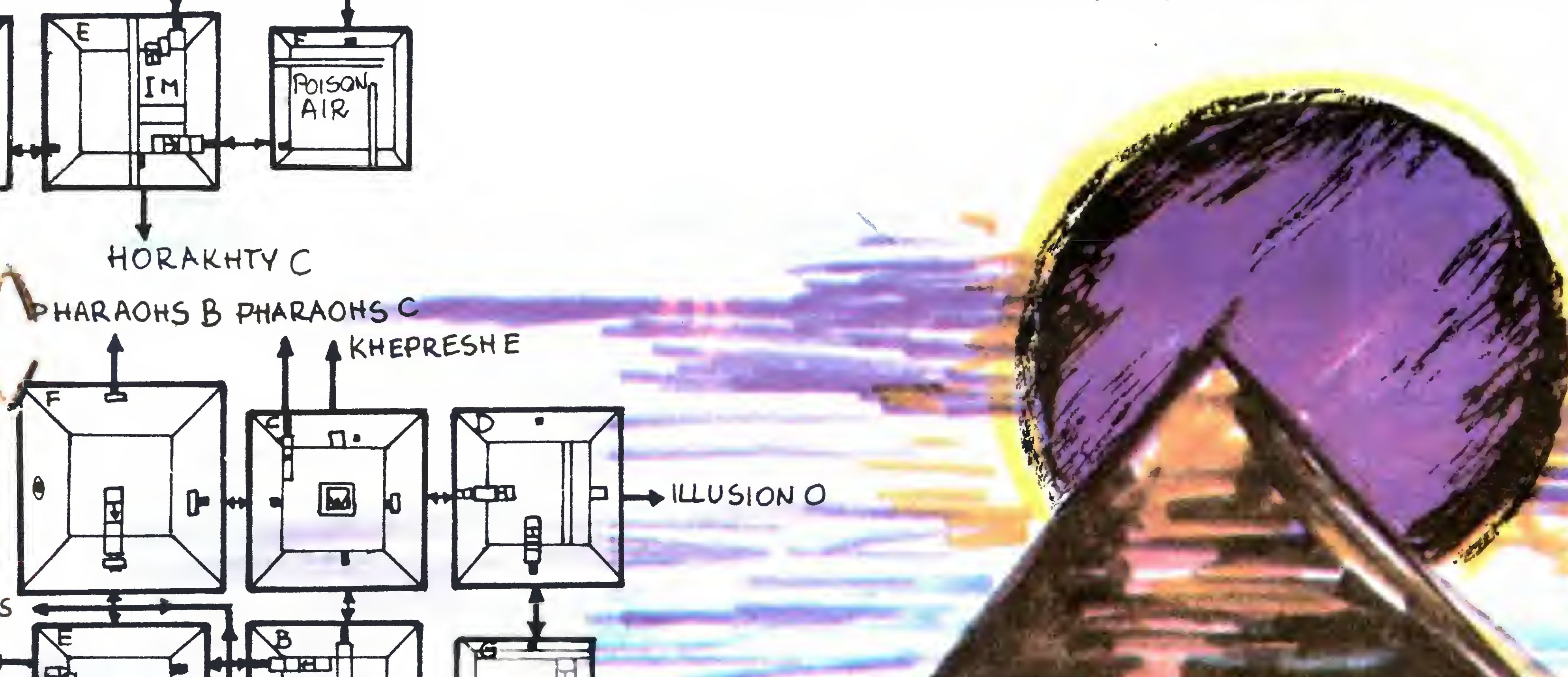
Możesz jednak wiedzieć, co powinien był robić Egon, by osiągnąć rezultat taki, jaki osiągnął. Poruszając się po wnętrzu piramidy uważaj, gdyż czyha na Ciebie wiele niebezpieczeństw. Piramida połączona jest z innymi, a gdzieś głęboko ukryta jest tajemnica, która za 24 godziny obróci Ziemię w pył. Twoim zadaniem jest zapobiec nieszczęściu. Jeśli Ci się nie powiedzie, nawet nie wstawaj jutro do szkoły.

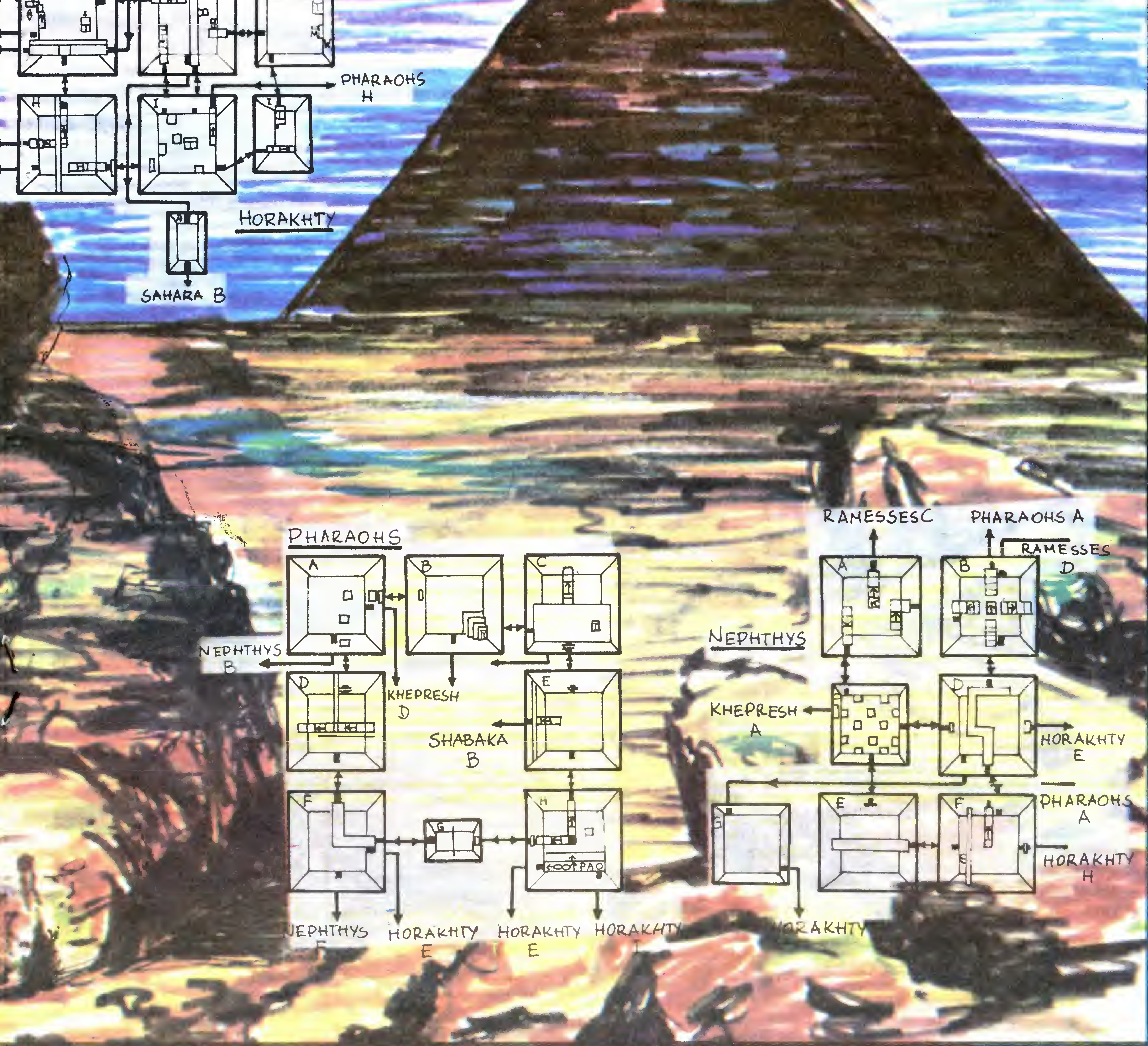
Firma: Incentive

Komputer: ZX Spectrum 48, Commodore 64

Gen







10 BAJTKOWA LISTA PRZEBOJÓW 10/89

Akcje policji stoją wysoko. Również armia, ta nie narzeka. Kosmici i karatecy trochę niżej, ale i tak przed komandosami, cegłami i śmigłowcami. Tak w skrócie i przenośni przedstawia się październikowe notowanie Listy Przebojów. Opanowały ją zdecydowanie gry mało „umysłowe", ale uwaga! Thunderbirds atakuje z drugiej dziesiątki! Nasz Gen Martinez ukończył wreszcie tę grę i właśnie kończy mapę. Do tego czasu Thunderbirds znajdzie się chyba wyżej, gdyż przymierza się do tego już od kilku miesięcy.

Na aktualne notowanie Listy nadeszło 2855 propozycji. Czytelnicy głosowali na 172 tytuły gier.

			ATARI	AMSTRAD	COMMODORE	SPECTRUM
1	ROBOCOP	↑	>	<	×	×
2	OPERATION WOLF	↓	>	>	×	×
3	RENEGADE III	!		<	×	×
4	CAPTAIN BLOOD		>	×	×	<
5	TRAZ	↑		<	<	<
6	CYBERNOID	↓	>		<	×
7	ROCKET RANGER	↓		<	<	
8	GAME OVER II	!	>	×	<	×
9	GUNSHIP	↓	>	×	×	<
10	TOTAL ECLIPSE	!			<	<



Kopertę nadesłał Łukasz Szafrański z Kielc.



BARBARIAN II

Conan wspiął się na mur i zeskoczył do ogrodu pogrążonego w cieniach, zagęszczonych czasami przez rozłożyste krzewy i poruszane wiatrem listowie. Żadne światło nie rozjaśniało okien zamczyska, odcinającego się czarną bryłą na tle nocnego nieba. Wojownik przemykał ostrożnie przez kolczaste chaszcze; w każdej chwili spodziewał się ujrzeć jakiegoś potwora. Strach przed śmiercią w zębach czegoś tak strasznego, że nawet sobie tego nie wyobrażał, nie powstrzymała go od dalszej wędrówki w stronę twierdzy. Mocniej ścisnął potężny miecz — jedyną broń, lecz jakże straszną w jego rękach. Nagle ujrzał przed sobą uchylone, masywne drzwi, za którymi widniał długi korytarz. Gdy przekroczył próg, wszystko zdawało się być przesiąknięte krwią — wrażenie to nie opuściło go jeszcze długo. Mroczny korytarz słabo rozświetlony był promykiem wnikającym obok zasłony przegradzającej odległe wyjście.

Grobowa cisza wisiała w powietrzu.

Conan jął się skradać korytarzem; zatrzymał się na końcu, by poprzez szparę w zasłonie ujrzeć oświetlony pokój o oknach szczelnie zasłoniętych aksamitnymi storami, tak by ani promyk światła nie przedarł się na zewnątrz. Komnata wydawała się pusta, lecz Conan instynktem barbarzyńcy północy wyczuł obecność żywej istoty. Nie chcąc zakończyć swego życia na tej kamiennej posadzce postanowił użyć fortelu.

Na środku pokoju stał bogato rzeźbiony mahoniowy stół, dźwigający bezlik naczyń z winem i kosztownymi potrawami. Barbarzyńca ujął w dłoń jeden z licznych zgaszonych świeczników zdobiących ścianę korytarza i cisnął nim w sam środek stołu. Hałas przewracających naczyń zmieszał się z szelestem siatki opadającej ze sklepienia i pokrywającej prawie całą komnatę. Po chwili zza jednej z licznych kotar wychyliła się dziwna postać, przypominająca trochę człowieka, lecz bardziej upodobniła się do wielkiej małpy. Jej bujne niebieskie futro, żywo kontrastowało z krwisto-czerwonymi wielkimi oczami.

Conan nie zastanawiał się długo. Dwa wielkie skoki, potężny zamach i potwór stracił głowę. Niestety, błąd jaki popełnił w tym momencie miał się srogo zemścić — Barbarzyńca zahaczył nogą o dźwignię zapadni. Okropny krzyk wydarł się spoza jego zsiniałych warg i nim zamarł, inny dźwięk zmieszał się z jego echem — łoskot ciała osuwającego się w głąb lochu...

Każdy z Was domyślił się zapewne, że znowu musicie wydostać Conana z opresji. Znalazł się on w dość dużym i przez to trudnym do przebycia labiryncie. Krążą po nim dostatecznie nieprzyjemne stwory, aby śmierć była dla Was zjawiskiem powszednim — zabijesz Ty albo on. Z ciekawszych postaci można wymienić: mocarzy z gigantycznymi maczugami, smoki zjadające głowy, skorpiony, kraby i jeszcze wiele innych. W podziemiach płyną też zatrute rzeki, niebezpieczne są także głębokie doły.

Miecz umieszczony poniżej ekranu wskazuje właściwy kierunek w którym należy iść. Niekiedy trzeba przejść przez grotę — czynisz to wciskając klawisz SPACE. Nie zapomnij także o zbieraniu magicznych amuletów oraz niezbyt sympatycznie wyglądających czaszek.

Pokonany poprzednio czarownik Drax zemści się na Tobie okrutnie, gdy dopadnie Cię w lochach swego zamczyska. I znowu walka na śmierć i życie rozstrzygnie, czy zagrasz w „Barbariana III"

Firma: Palace
Komputer: ZX Spectrum 48/+, Commodore 64, Atari ST.

LUKE

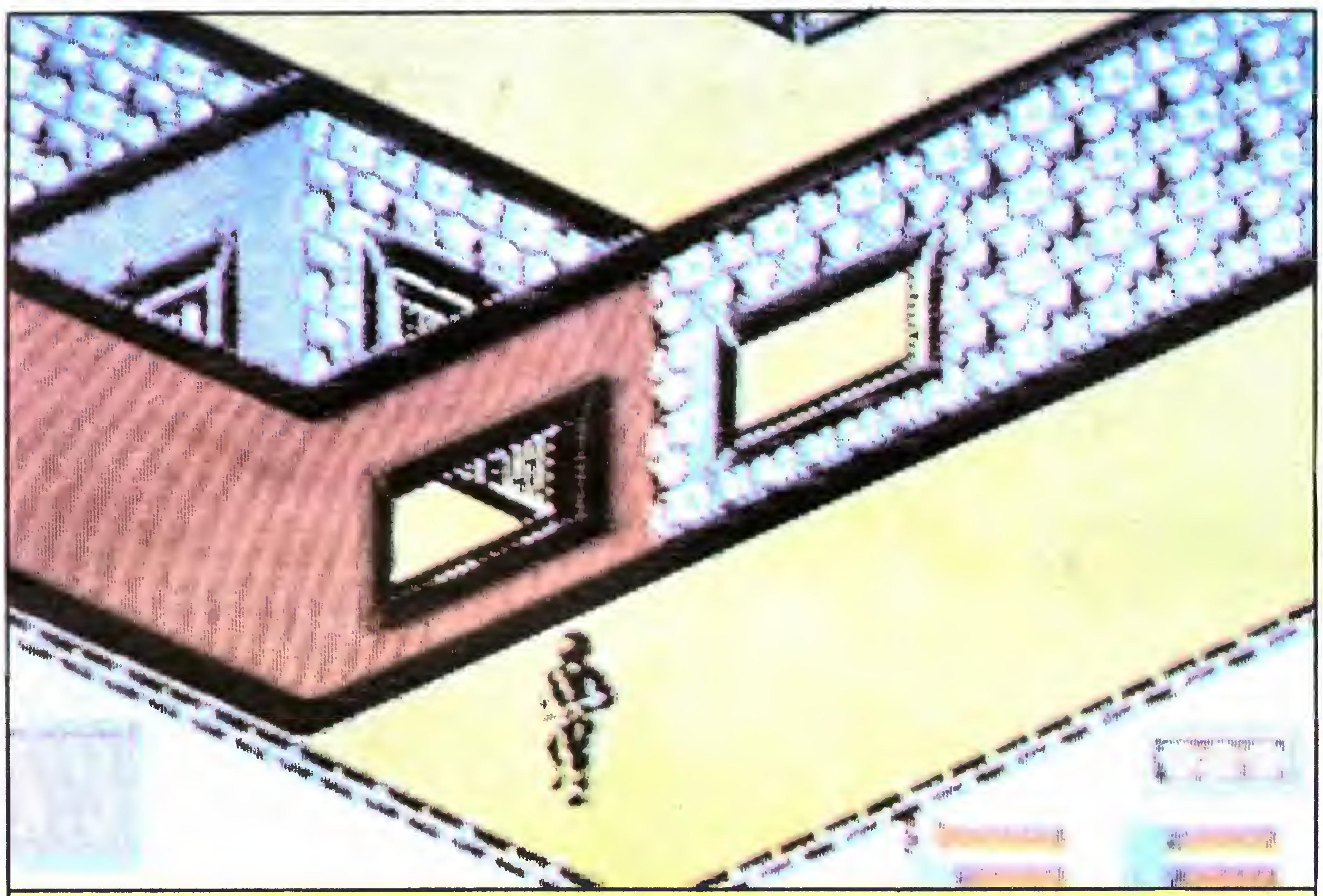
KRÓL I KRÓLOWA GIER



Wojtek Krasoń, lat 11.
V klasa Szkoły Podst. nr 44 we Wrocławiu.
Ulubiony komputer: ZX Spectrum.
Gry: Arkanoid, Galaxy.
Inne zainteresowania: filatelistyka.



Anna Grabaś, lat 12.
VI klasa Szkoły Podst. w Kłodzku.
Ulubiony komputer: Junior.
Gra: Arkanoid.
Hobby: książki.



STRIKE FORCE COBRA

Od kilku lat wszystkie państwa na świecie są zjednoczone. Skończyły się wojny, które powoli wyniszczały ludzkość. Wprowadzono też jeden system walutowy. I właśnie wtedy, gdy wszyscy myśleli, że odtąd nic już nie zakłóci spokoju Ziemian, ludzkość została wstrząśnięta tragiczną wiadomością.

Pewien szalony konstruktor, chcąc zawładnąć tym harmonijnym światem, zbudował olbrzymi superkomputer, który umieścił kilkanaście metrów pod powierzchnią Ziemi. Nad nim wybudował rozległy system korytarzy i pokoi. W niektórych pomieszczeniach porozstawiał terminale, za pomocą których sterowało się częściami komputera. Wejście do labiryntu było chronione przez różne urządzenia. Wykrycie intruza powodowało włączenie alarmu. Wewnątrz labiryntu umieszczono zdalnie kierowane roboty oraz działka, które włączają się automatycznie.

Niektóre drzwi można otworzyć mocnym kopnięciem, inne stając na płycie podnoszącej drzwi. Uważać trzeba też na strażników oraz na zapadnie, z których nie ma ucieczki. Jeżeli na swojej drodze napotkasz wysoki mur, przejdź go przy pomocy kolegi.

W pomieszczeniach z terminalami konstruktor umieścił dziewięciu profesorów-specjalistów, którzy pod przymusem obsługiwali superkomputer. Każdy z nich zna jedną literę szyfru, potrzebnego do zniszczenia superkomputera.

Bardzo trudno jest uwolnić naukowców z powodu olbrzymiej ilości alarmów i strażników. Oprócz tego labirynt ma kilka poziomów i poruszanie się po nim jest niezwykle trudne. Wiadomo też, że nikt jeszcze nie narysował mapy wszystkich poziomów.

Do akcji, której celem ma być uratowanie dziewięciu więźniów, wyznaczono ośmiu komandosów z całego świata. Niestety, tylko czterech z nich będzie brało udział w akcji. Musisz więc dokonać wyboru. Pomoże ci w tym teczka leżąca przed tobą i zawierająca dane osobowe wszystkich komandosów.

Imię i nazwisko: Joe Kowalski
Narodowość: Amerykanin
Stopień: sierżant
Miejsce i data urodzenia: Pensylwania, 1959
Waga: 82 kg
Wzrost: 165 cm
Dane dodatkowe: służył w formacji komandosów „GREEN BERET", obecnie członek DELTA FORCE. Bardzo dobrze walczy wręcz.
Specjalność: rozbrajanie ładunków wybuchowych.

Imię i nazwisko: Ester Stern
Narodowość: Żydówka
Stopień: kapitan
Miejsce i data urodzenia: Tel Awiw, 1961
Waga: 55 kg
Wzrost: 160 cm
Dane dodatkowe: w latach 1982–84 służyła w Izraelskiej Agencji Wywiadowczej.
Specjalność: taktyka walki w miastach i walka szturmowa.

Imię i nazwisko: Julio Fernandez
Narodowość: Hiszpan
Stopień: kapral
Miejsce i data urodzenia: Madryt, 1958
Waga: 76 kg
Wzrost: 170 cm
Dane dodatkowe: od 1979 pracował w Policji Hiszpańskiej, walcząc z terrorystami. Następnie został przeniesiony do wydziału walki z narkotykami.
Specjalność: taktyki włamywania się, niszczenie różnych systemów obronnych

Imię i nazwisko: Irina Viskowa
Narodowość: Rosjanka
Stopień: st. sierżant
Miejsce i data urodzenia: Leningrad, 1968
Waga: 60 kg
Wzrost: 170 cm
Dane dodatkowe: ukończyła specjalistyczną szkołę wojskową im. Suworowa w Moskwie. Uprawia gimnastykę olimpijską, mówi siedmioma językami.
Specjalność: elektroniczne systemy wczesnego ostrzegania.

Imię i nazwisko: Jan van Hensen
Narodowość: Holender
Stopień: major
Miejsce i data urodzenia: Utrecht, 1951
Waga: 91,5 kg
Wzrost: 187,5 cm
Dane dodatkowe: doskonały informatyk. Były komandos, obecnie służy w Brygadzie Antyterrorystycznej.
Specjalność: saper, ekspert do elektronicznych systemów rażenia.

Imię i nazwisko: Gerard du Pont
Narodowość: Francuz
Stopień: kapitan
Miejsce i data urodzenia: Algier, 1952
Waga: 83 kg
Wzrost: 177,5 cm
Dane dodatkowe: doświadczony informatyk, w 1978 pracował we Francuskiej Policji. Silny psychicznie, ale słaby w walce wręcz.
Specjalność: wysadzanie i minowanie.

Imię i nazwisko: Ross McWatt
Narodowość: Anglik
Stopień: sierżant
Miejsce i data urodzenia: Glasgow, 1957
Waga: 89 kg
Wzrost: 175 cm
Dane dodatkowe: w 1982 brał udział w konflikcie Falklandzkim.
Specjalność: techniki samoobrony i ratowania życia.

Imię i nazwisko: Ian Dawson
Narodowość: Anglik
Stopień: żołnierz
Miejsce i data urodzenia: Leeds, 1962
Waga: 90 kg
Wzrost 187,5 cm
Dane dodatkowe: dwukrotnie otrzymał medal za odwagę. Bardzo dobrze walczy wręcz.
Specjalność: prowadzenie działań wojennych w miastach.

Część czynności komandosów (rzut granatem, kopnięcie, itd.) trzeba obsługiwać z klawiatury.
Oto potrzebna lista:

ZX Spectrum:
1,2,3,4 — zmiana postaci, którymi sterujesz
E — lewo
D — prawo
I — góra
K — dół
W — podskok
A — skok z obrotem
V — strzelanie
C — kucanie
X — wstawanie
CAPS SHIFT — rzut granatem
SYMBOL SHIFT — kopnięcie

Commodore:
1,2,3,4 — zmiana postaci
5 — status misji
D — lewo
B — prawo
K — góra
M — dół
S — podskok
A — skok z obrotem
SPACE — strzelanie
– — kucanie
+ — wstawanie
E — rzut granatem
Q — kopnięcie

Komputer: ZX SPECTRUM 48/+, COMMODORE 64/128
Firma: Macmillan Software

LUKE

S.O.S.

Nie wiedzieć kiedy, z „S.O.S." zrobiła się „Pocztowa Giełda Programów". Większość ogłoszeń brzmi teraz jak oferta kupna-sprzedaży-wymiany programów. Nie potępiamy tylko tej ostatniej formy zdobywania programów. Spójrzcie na reportaż z Giełdy „Bajtka", to zastąpi Wam wyjaśnienia.

Póki co, drukujemy cały czas nadchodzące listy. Nie chcielibyśmy jednak, aby „S.O.S." pozostawało marketem dla cwaniaków, lecz jedynie prawdziwą giełdą wymiany informacji, doświadczeń i czasem programów między graczami.

Jestem posiadaczem ATARI 800XL. Mam kłopoty z grami Theatre Europe i Air Support. W Arkanoid doszedłem do 33 komnaty i nie wiem, co trzeba dalej robić.
Krystian Baran os. Wysokie 10/39, 31–819 Kraków

Posiadam ATARI 65XE. Poszukuję nieśmiertelności do gier: Arkanoid, Action Biker, Fort Apocalypse. Liczę na waszą pomoc.
Andrzej Szyk ul. Koźmińska 30, 61–417 Poznań

Pilnie poszukuję następujących gier na Atari w wersji kasetowej: Barbarian, Skate Crazy, Platoon, Mikie, Commando. Proponuję wymianę programów. Nie wiem też, jak wprowadzić nieśmiertelności przy użyciu Turbo-Copiera
Marcin Sulikowski ul. Traugutta 30/22, 66–400 Gorzów Wlkp.

Poszukuję nieśmiertelności do gier: Flying Shark, Tank 87, Raid Over Moscow na ZX Spectrum.
Ziemowit Sołyniuk ul. Metalowa 28a m 2, 80–744 Szczecin

Poszukuję następujących gier na ZX Spectrum: WEST BANK, TRASHMAN, SHOW JUMPING, BARBARIAN, MIKIE, THE ROCKY HORROR SHOW. W zamian oferuję inne lub zapłacę.
Rafał Piekarski ul. Żeromskiego 116/54, 26–600 Radom

Usilnie poszukuję szczegółowych opisów do gier: Heatrow, JetSet Willy, Zorro, Ace II, Conquest oraz ułatwień i POKE-ów do gier: Deathchase, Tutankhamun, GunFright, Thrusta na ZX Spectrum.
Damian Szołtysek ul. Radomska 11, 44–169 Gliwice 15

Proszę o nieśmiertelność do gier Commando, Bruce Lee, Exolon, Glug-Glug na ZX Spectrum oraz Boulder Dash na Atari
Bartosz Kowalski ul. Sikorskiego 13 m 30, 06–400 Ciechanów

Mam Atari 800XL i poszukuję miłych gier tekstowych, w rodzaju „The Dallas Quest", a także programu Coala Painter współpracującego z piórem świetlnym, w wersji na taśmę.
Tomasz Popik ul. Tarasy 7 m 4, 80–141 Gdansk

Poszukuję nieśmiertelności do gier: COMMANDO, ENDURO RACER, BOMB JACK oraz ALIENS (na „Spectrum").
Łukasz Szafrański Os. Barwinek 14/113, 25–113 Kielce

Pilnie poszukuję gry Robin Hood na Atari 65XE oraz dokładnego opisu do The Comet Game. Jak uruchomić gry Sargon II i Hacker? W zamian przyślę gry Darts, Jet Boot Jack, Cave Lord, Spitfire 40.
Paweł Osiński ul. Goszczyńskiego 7/80, 41–200 Sosnowiec



SZLACHETNE ZDROWIE

Jeśli bolą cię plecy, drętwieje szyja, pieką oczy i czujesz ból w nadgarstkach po pięciu godzinach spędzonych przed komputerem — jest to pierwszy sygnał ostrzegawczy.

Podobne objawy zmęczenia odczuwa regularnie wielu spośród 15 milionów Amerykanów spędzających całe dnie przy komputerowych terminalach. Sprawa stała się na tyle poważną, że zainteresował się nią amerykański magazyn U.S. NEWS ANS WORLD REPORT publikując na swych łamach artykuł Josepha Carey'a na ten temat.

My także otrzymujemy listy od naszych Czytelników, sygnalizujące pojawienie się problemów zdrowotnych jako konsekwencji nadmiernego używania komputera.

KONSEKWENCJE

„Od ponad pięciu lat interesuję się informatyką — pisze do nas czytelnik ukrywający się pod pseudonimem **„PREDATOR". — Z biegiem czasu stałem się maniakiem tworzenia programów oraz grania. Z początku przesiadywałem niekiedy 24 godziny przy klawiaturze. Zaliczałem kolejne gry, pochłonięty przez wspaniałe światy programów Enduro Racer, Bomb Jack czy Gryzor. Rzadko nie osiągałem wymarzonego celu jakim było ukończenie gry, a jeszcze rzadziej przebywałem na świeżym powietrzu. Bo przecież dzięki grom uzyskiwałem swoiste „odprężenie" i „odpoczynek".**

Podobna postawa spotykana jest często, zarówno na wschodzie jak i na zachodzie, wśród tzw. „maniaków komputerowych". Kilka lat temu na ekranach naszych kin prezentowany był film pt. „Gry wojenne". Jednym z jego bohaterów był genialny konstruktor, który odsunął się od świata symulując własną śmierć. W wydanej w roku 1986 antologii science — fiction pt. „Don Wolleim Proponuje" zawierającej najlepsze opowiadania z tego gatunku opublikowane w USA w roku poprzednim, bohaterowie „Naciśnij Enter" to genialni programiści ogarnięci szaleństwem.

Wbrew pozorom postacie te nie są obce rzeczywistości firm zajmujących się produkcją oprogramowania czy instytutów naukowych. Cienka jest granica dzieląca autentyczną pasję twórczą od szaleństwa, prowadzącego do zagłady. Taki los spotkał w przeszłości genialnego brytyjskiego matematyka i teoretyka budowy maszyn cyfrowych Alana Turinga. Obecnie też spotykamy „trudnych geniuszy" — twórca NeXT-a Steve Jobs słynie z okropnych manier. Niestety, coraz częściej można też natknąć się na ludzi chorych, zmęczonych, zdradzających objawy wyczerpania psychicznego, a nawet głębokich depresji prowadzących w chorobę.

Najpierw nauczyciele, a potem lekarze w USA, zauważyli wśród młodych ludzi korzystających z komputerów bardzo niepokojące objawy. Bardzo dobrzy uczniowie nagle opuszczali się w nauce, stawali się skryci, nie przejawiali ochoty do uczestnictwa w życiu klasy czy szkoły, niektórym gwałtownie pogorszył się wzrok.

Nikt na początku nie łączył tych problemów z komputerami. Jednak po wnikliwym dochodzeniu okazało się, że winne są właśnie one, a raczej nowa pasja uczniów.

„**Ogarniał mnie dreszczyk emocji** — pisze „**PREDATOR**" — **gdy widziałem przed sobą kolejną część starego ulubionego przeboju. To wszystko prowadziło do strasznego końca. Moje wyniki w szkole zaczęły gwałtownie spadać i gdyby nie interwencja rodziców nie wiem jak by się to skończyło. Przez pewien czas nie miałem styczności z moją ulubioną zabawką, dużym nakładem pracy nadrobiłem zaległości w szkole i wszystko powoli wracało do normy. Niestety, były to tylko złudne nadzieje. Coś niedobrego zaczęło się dziać z moim zdrowiem, pojawiły się bóle głowy oraz oczu.**

Myślałem, że jest to spowodowane przemęczeniem. Niestety, mój wzrok uległ znacznemu pogorszeniu. Wizyta u okulisty była dla mnie „kubłem zimnej wody". Musiałem zacząć nosić okulary."

JAK SIĘ BRONIĆ

Jeśli nie możesz zdecydować się na zerwanie ze swoją pasją, skorzystaj z rad zamieszczonych przez Josepha Carey'a w U.S. NEWS AND WORLD REPORT:

Po pierwsze — przy pracy z komputerem należy unikać wszelkich ruchów naprężających mięśnie. **Zbyt wysokie krzesło może spowodować opuchnięcie kostek u nóg**, gdyż uda zbyt mocno naciskają na siedzenie, a to utrudnia właściwy przepływ krwi w nogach. **Klawiatura, której kąt nachylenia nie jest dostosowany do nadgarstków, może wywołać tzw. syndrom przewodu nadgarstkowego**; jest to schorzenie spowodowane zapaleniem ścięgien nadgarstkowych, w konsekwencji czego następuje podrażnienie nerwów. W idealnych warunkach pracy stawy kończynowe powinny znajdować się w pobliżu środka przestrzeni, w jakiej się mogą poruszać, by ograniczyć zbędne ruchy pracownika do minimum. **Krzesła powinny być tak dopasowane, by stopy mocno opierały się o podłogę, a łydki znajdowały się w pionie.**

Klawiaturę należy umieścić w ten sposób aby ramiona były pionowo a przedramiona poziomo. Linia wzroku w stosunku do ekranu powinna przebiegać pod kątem 10–20 stopni poniżej linii horyzontalnej terminalu. Większy kąt może spowodować drętwienie szyi i barków.

Wskazane są okresowe, 5-minutowe przerwy na gimnastykę. Carey opisuje eksperyment wprowadzony w oddziale komputerowego nadzoru firmy Pacific Gas and Electric Company w San Francisco, gdzie zainicjowano program krótkich ćwiczeń gimnastycznych dla pracowników. Po trzech miesiącach usprawniona została komputerowa obsługa depozytów firmy, co dało miesięcznie 1,2 miliona dolarów z tytułu dodatkowych wpływów od odsetek.

Znacznie trudniej jest zmniejszyć zmęczenie oczu wywołane nieustanną obserwacją ekranu. Szczególnie niebezpieczne jest występowanie odbić światła pochodzącego z innych źródeł np. lamp oświetleniowych czy słońca. W USA wyprodukowano specjalne filtry przeciwmigotliwe, które, założone na ekran, rozpraszają zbędne odblaski. Istotne może też być samo ustawienie komputera — w taki sposób, aby została zmniejszona do minimum możliwość wystąpienia tego zjawiska.

Jeśli jednak wystąpią zaburzenia ostrości widzenia, bóle głowy i objawy zmęczenia, należy niezwłocznie poddać się badaniom lekarskim.

Bardzo wiele obaw budzi też występowanie szkodliwego dla zdrowia promieniowania monitora. Zalecane jest ustawienie go tyłem do ściany. **Stanowczo zabrania się ustawiania stanowisk pracy w szeregu jedno za drugim.** Można je ustawiać obok siebie, ewentualnie łącząc po dwa przedzielone ścianką ze specjalnego materiału pochłaniającego niekorzystne promieniowanie.

Coraz częściej **zabrania się też pracy przy komputerach kobietom ciężarnym ze względu na niebezpieczeństwo poronienia.** Co prawda nie udowodniono bezpośredniego wpływu promieniowania monitorów na płód, ale ostrożność nie zawadzi. Pamiętajmy, że sam komputer nie jest źródłem promieniowania. Być może dlatego wiele obiecują sobie producenci ekranów ciekłokrystalicznych — są one o wiele bezpieczniejsze pod względem radiologicznym od klasycznych monitorów. Niestety, jak na razie ich wadą jest zbyt mało kontrastowy obraz, co wpływa niekorzystnie na wzrok.

W krajach wysoko rozwiniętych coraz większe sumy wydaje się na właściwą organizację miejsca pracy, rozumiejąc, że zdrowy i wypoczęty pracownik przynosi większe zyski firmie i w efekcie początkowy wydatek zwraca się z nawiązką. **Do Polski dotarły już specjalne siatki zakładane na monitor, warto je polecić tym, którzy pracują przy komputerach.** Pozostałym zalecamy rozsądek i umiar w korzystaniu z ulubionej zabawki.

●

Już starożytni wiedzieli, że nadmierne korzystanie z przyjemności staje się w końcu udręką. Wielogodzinne katowanie joysticka przy grze Overkill, w dokładnie zaciemnionym pokoju, aby rodzice nie zauważyli, do niczego dobrego nie prowadzi. A jeśli lekarz stwierdzi, że zdrowie jest w niebezpieczeństwie to pozostaje jedynie wyjście opisane przez „Predatora".

„**Sprzedałem komputer i rozpocząłem normalne życie (bieganie, pływanie, wycieczki). Jedyna rzecz, która została po ulubionej pasji, to dobra znajomość języka angielskiego, znakomity refleks i... okulary. Moja styczność z grami ogranicza się teraz tylko do czytania opisów i oglądania map**".

Marek Czarkowski

WORKSTATION

czyli „coś większego"

Siedem lat temu kupiłem swój pierwszy mikrokomputer — zwierza się Czytelnikom lutowego numeru amerykańskiego miesięcznika „BYTE" Bill Nicholls — to był Radio Shack Model 3, który posiadał tylko 16 Kb pamięci. Trzy lata później kupiłem Sanyo 555. Miał on 128 Kb pamięci i dwie jednostronne stacje dysków. Moją kolejną maszynką był „klon" IBM PC XT z mikroprocesorem pracującym z częstotliwością 8 MHz, ale nie byłem z niego zadowolony. Obecnie mam „coś większego" — jest to ALR 386/220 z 32 bitowym mikroprocesorem pracującym z częstotliwością 20 MHz. Jego system operacyjny to Unix V.3.Q, do tego monitor wysokiej rozdzielczości z kartą VGA i właściwie można pracować!

Bill Nicholls ma rację pisząc, że „można pracować", bowiem zestaw, w jaki się zaopatrzył, właściwie tylko do tego służy. Bill jest właścicielem workstation, czyli stanowiska roboczego, albo stacji roboczej, jak kto woli.

Co prawda, Czytelniku, jeśli jesteś posiadaczem Spectrum, Atari czy Commodore możesz dojść do wniosku, że tekst ten jest dla Ciebie mało ważny. Liczę jednak na to, że przeczytasz go z zainteresowaniem. Nawet właściciel „Syrenki" zwanej „Skarpetą" lubi czasem sięgnąć po artykuł o zaletach Rolls-Royce'a.



Foto 1. Apollo DN3500. 4 miliony operacji na sekundę

KRÓTKA HISTORIA STACJI ROBOCZYCH

Wszystko zaczęło się w roku 1970. Pierwsze stacje miały monitory o wyższej rozdzielczości, pozwalały na pracę w sieci z minikomputerem i były stosowane przez inżynierów i projektantów. Kosztowały kilka tysięcy dolarów — w połączeniu z minikomputerem kilkanaście. Przykładem może tu być **VAX–11/780** firmy **Digital Equipment Corp**. Współczesne stacje robocze posiadają 19calowe monitory o rozdzielczości ponad 1000x800 punktów, digitizer i profesjonalną klawiaturę z myszką. Są niezastąpione przy projektowaniu, modelowaniu i symulacji całych nawet systemów mechanicznych. A co najważniejsze, ich cena stale spada.

W roku 1982 pierwsze IBM PC z podwójną stacją dysków i zwykłym monitorem monochromatycznym kosztowały około 5000 tysięcy dolarów. Dziś za tę sumę możesz kupić komputer z 32 bitowym mikroprocesorem, z pamięcią RAM o pojemności 2 Mb, twardym dyskiem i monitorem wysokiej rozdzielczości! Zapotrzebowanie na tego typu sprzęt od dawna zgłaszali naukowcy i projektanci. Komputerowe wspomaganie prac badawczych i konstrukcyjnych wymaga sprzętu doskonalszego niż popularne „pecety". Potrzebna jest duża szybkość obliczeń, której komputery osobiste nie posiadają, i co równie istotne, możliwość pracy w sieci z dużymi komputerami.

Już na starcie stacje robocze były lepsze. Budowano je w oparciu o 16 bitowe mikroprocesory, podczas gdy PC zaczynały z ośmiobitowymi Intelami. Dziś stosowane są 32 bitowe mikroprocesory produkowane przez firmy **Intel** i **Motorola**. Dodatkowo wspomagają je koprocesory arytmetyczne firm **Weitek** lub **Intel**, co znacznie podnosi prędkość wykonywania operacji.

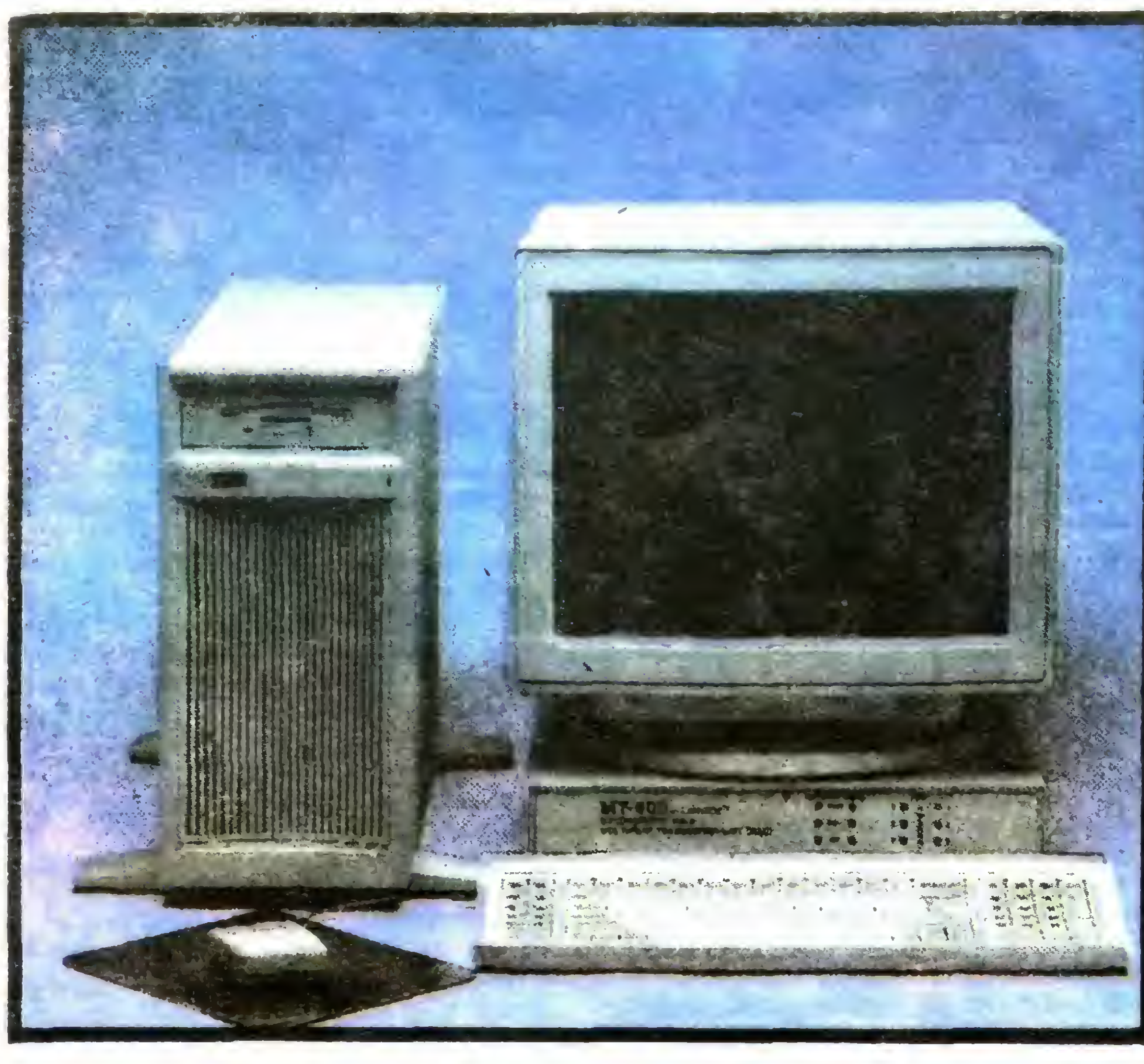


Foto 2. Sun 386i/150. 3,5 miliona operacji na sekundę

APOLLO, SUN, NEXT

Wśród producentów stacji roboczych liczy się tylko kilka firm. Pierwszą z nich jest założona w roku 1980 w Chelmsford **Apollo Computer Inc**. Ich pierwsza stacja z tego roku była oznaczona symbolem DN 100. Konstrukcja sprzętu bazowała na mikroprocesorze **Motorola 68000**, a cena tego „cuda" wynosiła... 60 tysięcy dolarów! Najnowszy model oznaczony symbolem **Apollo DN4500** wyposażony w twardy dysk o pojemności 155 MB minimum (jest możliwość zastosowania dysków o pojemności 696 MB!), z monitorem o rozdzielczości 1280x1240, i szybkości przetwarzania równej 7 MIPS kosztuje w USA 18 990 dolarów. Systemem operacyjnym komputera jest **Aegis**, **Unix SV** albo **Unix BSD** — zgodnie z życzeniem kupującego. Firma zadbała też o oprogramowanie sieciowe pod nazwą **Domain/OS**. Oczywiście dwa pozostałe produkowane aktualnie modele **Apollo DN3000** i **Apollo DN3500** mają możliwość współpracy zarówno z najnowszą konstrukcją, jak i jej oprogramowaniem, co stanowi zachętę dla stałych klientów. Dwa lata po powstaniu **Apollo** w miejscowości Mountain View rozpoczęła działalność firma **Sun Microsystems Inc**. To właśnie jej zawdzięczamy próbę ustalenia dwóch podstawowych standardów wśród stacji roboczych — systemu operacyjnego i oprogramowania sieciowego — systemu operacyjnego **Unix** i sieci **Ethernet**. Od września 1985 roku **Sun** systematycznie wprowadzał na rynek nowe serie swych stacji: **Sun–3/160**, następnie **Sun–3/260** w roku 1986, **Sun–4/260** w roku 1987, oraz **Sun–4/110** i **Sun3861/150** (na zdjęciu) w 1988 roku. Możliwości tych komputerów są podobne do wcześniej omawianych produktów firmy **Apollo**. Ciekawostką jest możliwość zastosowania popularnego wśród komputerów osobistych **IBM PC** systemu operacyjnego **MS-DOS** obok firmowego **Unix** występującego pod nazwą **SunOS**. Specjaliści firmy podkreślają, że — *to sieć jest komputerem*. Może mieć to znaczenie dla użytkowników planujących zorganizowanie pracy w systemie sieciowym wskazując na tych, którzy mają doświadczenie w budowie tego typu instalacji. W chwili obecnej **Sun** i **Apollo** prowadzą między sobą zaciętą walkę konkurencyjną, do której próbuje się włączyć inni.

Jedną z takich firm jest **Silicon Graphics Inc.** istniejąca od roku 1984 i mająca kwaterę główną w tej samej miejscowości co **Sun** — Mountain Viev. Nic mi nie wiadomo o



Foto 3. SGI Personal Iris. 10 milionów operacji na sekundę



Foto 4. NeXT. 4 miliony operacji na sekundę

tym, aby pracownicy obu tych firm przebijali opony w samochodach konkurencji.

Silicon Graphics Inc. specjalizuje się w produkcji stacji o wspaniałych możliwościach graficznych. „Okrętem flagowym" firmy jest komputer o nazwie **SGI Personal Iris** (na zdjęciu). Głównymi odbiorcami są projektanci, dla których kolor ma ogromne znaczenie. **Iris** podobno jest w tej chwili najlepsza pod tym względem. Jej systemem operacyjnym jest oczywiście **Unix**, nie wiedzieć czemu występujący pod nazwą **IRIX**. Firma zadbała też o wyposażenie komputera w pakiet procedur graficznych. Podobno **Flight Simulator** prezentowany na monitorze tej stacji jest czymś absolutnie doskonałym.

W końcu ubiegłego roku ogromną sensacją w komputerowym światku stała się prezentacja najnowszego „dziecka" współtwórcy **„Apple" Steve Jobsa** — komputera **NeXT**. Sam **Jobs** należy już mimo młodego wieku do legendy mikrobiznesu. Pierwszy milion dolarów zarobił po dwudziestce. Kiedy odchodził z firmy **Apple Inc.** był już bardzo bogatym człowiekiem. Rozstanie nie należało do najprzyjemniejszych, tym bardziej, że firma po klęskach finansowych spowodowanych nie zawsze trafnymi decyzjami **Jobsa** znalazła się w poważnych tarapatach. Być może dlatego nikt nie dawał mu zbyt wielkich szans na powrót do czołówki konstruktorów Krzemowej Doliny.

Jobs na odchodnym powiedział dziennikarzom — *Ja jeszcze wrócę!* I słowa dotrzymał. Prezentacja **NeXT-a** w wielkiej sali Davies Symphony Hall w San Francisco stała się wielkim show w iście hollywodzkim stylu. Bilety wstępu kosztowały u „koników" 500 dolarów, jednak ci którym udało się je dostać zgodnie twierdzili, że warto było.

Trzy godziny prezentował **Jobs** oniemiałej z wrażenia publiczności możliwości swojego komputera. Grał na nim jak na organach, demonstrował trójwymiarowe obrazy molekuł, sprawiał, że maszyna zamieniała się w wysokiej klasy odtwarzacz płyt kompaktowych.

NeXT natychmiast został zaliczony przez znawców przedmiotu do tzw. workstation, prawdopodobnie wbrew intencjom twórcy. **Jobs** ma zamiar sprzedawać swój produkt szkołom i uniwersytetom amerykańskim. Cena **NeXT-a** została ustalona na bardzo niskim poziomie 6500 dolarów, co sprawia, że w chwili obecnej praktycznie nie ma on konkurencji.

Twórcy tego komputera zastosowali szereg nowatorskich rozwiązań technicznych, z których najciekawszym wydaje się zastosowanie kasowalnych magnetyczno-laserowych dysków produkcji firmy **Canon** o pojemności 256 MB. Wbrew pozorom nie jest to dużo. Wspomniany wcześniej komputer **SGI Personal Iris** może dysponować pamięcią **2000 MB**! Nietypowy jest też monitor **NeXT-a** — zaledwie 17 cali i w dodatku czarno-biały. Pytany o to **Jobs** odparł, że chodziło o minimalizację kosztów i komfort pracy. *Ostatecznie NeXT nie służy do gier* — skomentował problem jeden z dziennikarzy obecnych na pokazie.

NeXT został zbudowany w oparciu o 32 bitowy mikroprocesor **Motorola 68030** pracujący z częstotliwością **25 MHz**. Razem z koprocesorem arytmetycznym tej samej firmy moc przetwarzania **NeXT-a** wynosi **4 MIPS** — cztery miliony instrukcji na sekundę, co też nie jest rekordem — **SGI Personal Iris** osiąga **10 MIPS**, ale jednocześnie jest prawie trzykrotnie droższy — kosztuje 16 000 dolarów.

Wydaje się, że przed **NeXT-em** jest duża przyszłość. Nazwisko Jobsa jeszcze wiele znaczy a i cena komputera jest bardzo zachęcająca. Koncern **IBM** podpisał już umowę z **NeXT Inc.** o dostosowaniu systemów oprogramowania obu firm, co może wróżyć długie panowanie **„Następnego"** na rynku amerykańskim w latach dziewięćdziesiątych.

PIĘTA ACHILLESA

To oczywiście oprogramowanie. Tysiące programów użytkowych, gier itp., które napisano dla komputerów ośmio- i szesnastobitowych, wymagało blisko dziesięciu lat pracy sztabów programistów i indywidualnych zapaleńców. Nawet dziś znajdą się tacy, którzy potrafią zaskoczyć użytkowników nowym pomysłem na stary sprzęt. Postęp w dziedzinie sprzętu wyprzedza znacznie postęp w oprogramowaniu. Można przyjąć, że obecnie dobrze oprogramowane są komputery zbudowane w oparciu o szesnastobitowe mikroprocesory. Stacje robocze mimo ogromnym możliwości nie do końca są wykorzystywane.

Standardowym systemem operacyjnym jest **Unix**, którego podstawową zaletą jest możliwość jednoczesnej obsługi wielu użytkowników, a także możliwość realizacji wielu zadań jednego użytkownika. Poza tym bez większych problemów można go zaadoptować na różnych zestawach komputerowych, a także łatwo można tworzyć programy pomocnicze, dzięki czemu system szybko się rozrasta. System **Unix** został napisany w języku **C** i jest kilka razy większy od np. **MS-DOS-a**. Nie jest on też prosty w użyciu, być może dlatego, że twórcy **Unix-a** przyjęli założenie iż korzystać z niego będą jedynie specjaliści. Bez względu jednak na problemy jest on dziś standardem.

Programów użytkowych napisanych z myślą o workstation jest w porównaniu z „pecetami" niewiele. Firma **Apollo Computer Inc.** podaje w swym katalogu listę 925 produktów software-owych. **Sun Microsystems Inc.** umieściła na swej liście 1200 propozycji, a **Silicon Graphics** jedynie 75. **NeXT Inc.** liczy prawdopodobnie na aktywność firm softwarowych i użytkowników. Oprogramowanie **„Kostki"** — bo taka też nazwa przyjęła się w USA, ogranicza się między innymi do kompilatora języka C, edytora tekstu, słownika, tezaurusa, programu muzycznego, kompletnej edycji dzieł Szekspira itp. Nie jest to wiele zważywszy, że producenci software'u wręcz rozpieszczają swoich klientów. Oczywiście ta sytuacja ulegnie zmianie.

Postęp w dziedzinie sprzętu i oprogramowania, tak szybki jeszcze pięć lat temu, uległ jakby wyhamowaniu. Kilka projektów, z którymi wiązano nadzieje, nie zostało zrealizowanych np. skonstruowanie komputera V generacji, czy program komputera Abaq firmy Atari. Pojawienie się serii **PS/2** firmy **IBM** było próbą wprowadzenia nowego standardu w dziedzinie komputerów osobistych, na ile udaną — czas pokaże.

System	Mikroprocesor	MIPS	Pamięć RAM	Monitor	Dysk	Cena
Apollo DN3000	Motorola 68020 - 16	1,7	2-8 MB	1024×800	0-696 MB	4990 $
Sun-3/50	Motorola 68020 - 15	1,5	4 MB	1152×900	71-1300 MB	4995 $
NeXT	Motorola 68030 - 25	4	8-16 MB	1120×832	256-926 MB	6500 $
ALR 386/220	Intel 80386 - 20	3,5	1-10 MB	640×480	67-320 MB	7695 $
Sun 386i/150	Intel 80386 - 20	3,5	4-16 MB	1024×768	91-981 MB	7990 $
Sun 3/60	Motorola 68020 - 20	3	4-24 MB	1152×900	71-1300 MB	8900 $
Apollo DN3500	Motorola 68030 - 25	4	8-32 MB	1024×800	0-696 MB	8990 $
Sun 386i/250	Intel 80386 -	5	4-16 MB	1024×768	91-981 MB	11990 $
SGI Personal Iris	R2000A - 12	10	8-16 MB	1280×1024	155-2000 MB	16000 $
Sun-4/110	SPARC - 14	7	8-32 MB	1152×900	141-1300 MB	18900 $
Apollo DN4500	Motorola 68030 - 33	7	8-32 MB	1280-1024	155-696 MB	18990 $

Opracowane na podstawie miesięcznika "Byte". Ceny dotyczą podstawowej konfiguracji sprzętu.
MIPS - miliony operacji na sekundę

Personal workstations mają przed sobą przyszłość, tym bardziej, że dzięki **NeXT-owi** dostały się one do klas i sal wykładowych. Jeśli **NeXT** stanie się standardem — a szanse są ogromne, ponieważ oświata w USA ma pieniądze — pojawi się grono młodych ludzi, przyzwyczajonych do nowej jakości w dziedzinie sprzętu komputerowego. Oni mogą utrwalić pozycję produktu na rynku. Stacje robocze są też jakby poligonem doświadczalnym dla konstruktorów komputerów osobistych przyszłości. To najpierw na nich eksperymentuje się z najnowszymi rozwiązaniami technologicznymi, takimi jak bardzo szybkie mikroprocesory o ograniczonej liczbie rozkazów, wykonane w technologii RISC, czy koprocesory arytmetyczne. Ponieważ stacji nie buduje się w dużych seriach, a ich cena jest wysoka, producenci mogą pozwolić sobie na śmielsze pomysły. Najlepsze rozwiązania zostana zastosowane w seryjnie produkowanych za pięć, sześć lat komputerach osobistych. Stacje robocze — to prawdziwa Formuła I współczesnego komputerowego interesu.

Nas te problemy nie dotyczą, polskie szkolnictwo boryka się z brakiem środków i ogólną niemożnością. Rodzice jak mogą starają się zapewnić swym pociechom kawałek klawiatury ale ja jestem przekonany, że doczekam się czasów, gdy we wstępie artykułu będę mógł napisać:

Od kilku lat miałem IBM PC ale dziś kupiłem „coś większego" — ostatni produkt ELWRO i koncernu Elpol — komputer The Junior Personal DGB 56000. Czterdzieści transputerów na pokładzie! moc obliczeniowa rzędu 100 MIPS. A na dodatek emulator „Gumiaka" na dwudziestocalowym kolorowym monitorze!

Marek Czarkowski

źrł. „BYTE", „Personal Computer World"

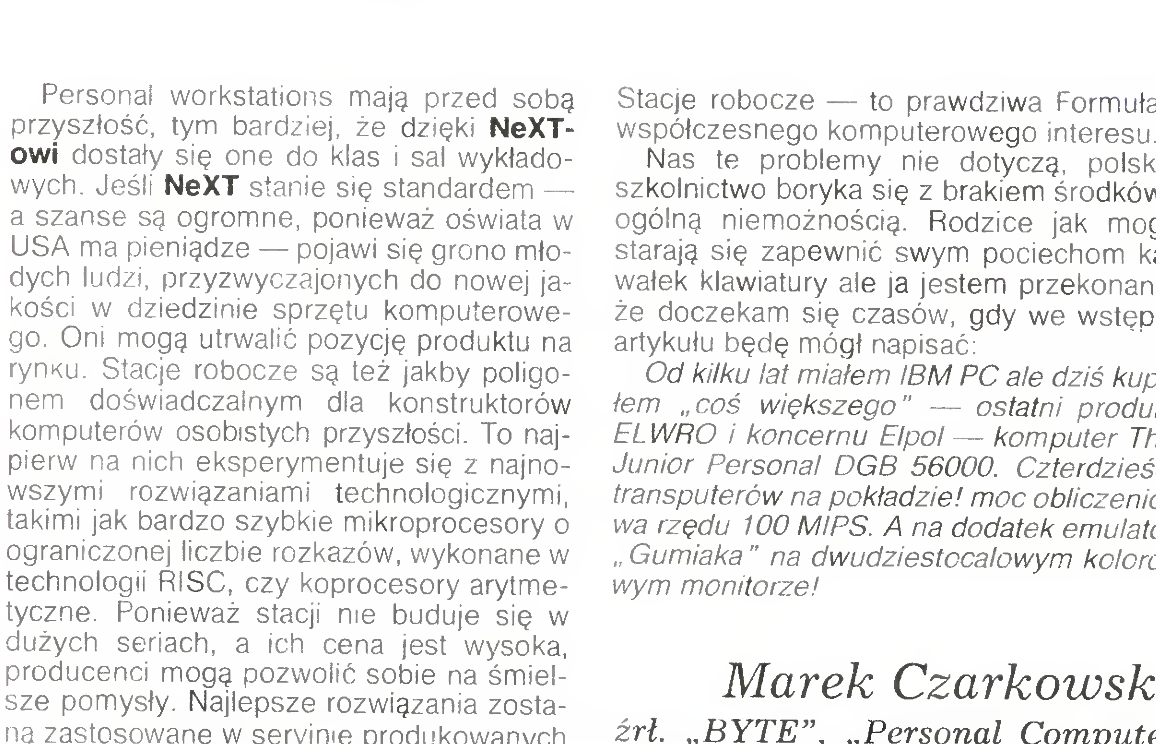


Foto 5. Projekt dziecięcego rowerka na ekranie komputera SGI Personal Iris

Amigowisko

Sobota 30 września 1989 r.

Krakowska Akademia Górniczo Hutnicza, blok b3. Wchodzimy. Na drzwiach kartka: „Witamy uczestników IV ogólnopolskiego zjazdu użytkowników Amigi. Ostatnie piętro, dojazd windą w głębi holu." Już w windzie słychać znajome dźwięki „Gliniarza z Beverly Hills" — chyba trafiliśmy. Na lewo sala, z której dolatuje mieszanina dźwięków, jakie tylko mogą wydać Amigi. Po prawej stół z magnetowidem i dwóch ludzi z zarządu klubu ACC — organizatora imprezy. Po opłaceniu składki (1500 zł) wchodzimy do sali z komputerami. Około trzydziestu Amig i dwustu ludzi pracuje na pełnych obrotach. Wszyscy kopiują programy, z rzadka ktoś ogląda nowy nabytek albo oferowany mu towar. Od prezesa ACC Piotra Podremańskiego dowiaduje się, że zloty Amigantów organizowane są mniej więcej raz na trzy miesiące. Nie ma określonego miejsca zjazdów — zależy gdzie uda się wynająć salę. O zlotach członkowie klubu zawiadamiani są na trzy tygodnie przed imprezą. Zloty te są w zasadzie jedynymi zebraniami klubu ACC. Dla członków z poza Krakowa trzy miesiące to termin optymalny, a w Krakowie — nowe programy rozchodzą się w jeden dzień, więc nikomu nie chce się organizować zebrań. Na zlocie można zapisać się do klubu, dostać aktualny spis członków i kupić biuletyn (wydawany oczywiście przy pomocy DTP na Amidze). Po uzyskaniu tych informacji rzuciliśmy się do zbierania „materiałów dziennikarskich" tzn. zaczęliśmy wymieniać programy. Gry i programy demonstracyjne wabiły muzyką i grafiką. To nieprawdopodobne, co można wycisnąć z Amigi. Na przykład „Lois Lois" Criter'ów to ponad dwie minuty muzyki i śpiewu stereo. Przebojami giełdy były „Dragons Lair" — bardziej film animowany niż gra i „Walker Demo" — filmik animowany z maszyną kroczącą z „Imperium kontratakuje" w roli głównej. Niestety oba programy dla posiadaczy co najmniej 1MB pamięci. Zwykłych śmiertelników bawiły „Grand slam monsters" „Wind surf Willy", „Barbarian II", „Test drive II", „Robocop", „Indiana Jones and the last crusaide" i wiele innych gier z 1989 roku. Pojawiły się też „IBM transformer 1.33" — szybszy od 1.21. „Foton painter II", „Reflection" — program do tworzenia rysunków realistycznych uwzględniających fakturę czy przeźroczystość przedmiotu. „Sound tracker 2.0" — najlepszy chyba program muzyczny i dziesięć dysków z instrumentami muzycznymi do niego. Za kilka programów można było dostać podkładkę do myszy, a jeśli kogoś zmęczyło bieganie z dyskietkami od komputera do komputera, mógł wyjść na chwilę z sali, zjeść kanapkę i obejrzeć jakiś film z magnetowidu. Jedni kopiowali programy spokojnie i z rozwagą, oglądając najpierw towar, inni kopiowali wszystko, co wpadło w rękę.

Około czwartej po południu towarzystwo zaczęło powoli zbierać dyskietki rozrzucone po wielu stanowiskach, składać komputery i rozchodzić do domów. Tam obejrzy się nowe nabytki, usunie wirusy z dysków, uporządkuje dyskietki. Następne „copy party" za trzy miesiące...

materiały zebrali:
Marcin Bójko i Mateusz Krauze

Adres klubu:
Piotr Podremański
31-867 KRAKÓW
oś II Pułku Lotniczego 10/40
tel. 47-46-80

SMOKOLOGIA

```
  50 DATA "        "
  60 DATA "        "
  70 DATA "        "
  80 DATA "        "
  90 DATA "        "
 100 DATA "        "
 110 DATA "        "
 120 DATA "        "
 130 DATA "        "
 140 DATA "        "
 150 DATA "        "
 160
 500 BORDER VAL "2": PAPER VAL "
2": CLS : GO SUB VAL "1e3": REST
ORE : FOR f=VAL "2" TO VAL "14":
 PRINT AT f,VAL "2"; PAPER VAL "
4";"                  ": NEXT f
 510 INK VAL "7": PLOT VAL "14",
VAL "161": DRAW VAL "83",NOT PI:
 DRAW NOT PI,VAL "-107": DRAW VA
L "-83",NOT PI: DRAW NOT PI,VAL
"107": INK NOT PI
 520 FOR f=VAL "3" TO VAL "13":
READ a$: PRINT PAPER VAL "4"; IN
K SGN PI;AT f,VAL "3";a$: NEXT f
 530 POKE VAL "23729",SGN PI
 550 LET y=VAL "3"
 560 PRINT AT y,VAL "13";"AB"
 570 IF IN VAL "31"=VAL "4" AND
y<VAL "13" THEN LET y=y+SGN PI:
GO TO VAL "700"
 580 IF IN VAL "31"=VAL "8" AND
y>VAL "3" THEN LET y=y-SGN PI: G
O TO VAL "800"
 590 IF IN VAL "31"=VAL "16" THE
N RESTORE : FOR f=SGN PI TO (y-V
AL "2"): READ a$: NEXT f: PRINT
AT VAL "13",NOT PI;: LOAD *a$
 600 GO TO VAL "560"
 700 PRINT AT y-SGN PI,VAL "13";
"  ";AT y,VAL "13";"AB": BEEP VA
L ".001",CODE "A": GO TO VAL "56
0"
 800 PRINT AT y+SGN PI,VAL "13";
"  ";AT y,VAL "13";"AB": BEEP VA
L ".001",CODE "A": GO TO VAL "56
0"
1000 RESTORE VAL "1e3": FOR f=NO
T PI TO VAL "7": READ s: POKE US
R "a"+f,s: NEXT f
1010 DATA VAL "16",VAL "48",VAL
"127",VAL "255",VAL "127",VAL "4
8",VAL "16",NOT PI
1020 FOR f=NOT PI TO VAL "7": RE
AD s: POKE USR "b"+f,s: NEXT f
1030 DATA NOT PI,NOT PI,VAL "252
",VAL "252",VAL "252",NOT PI,NOT
 PI,NOT PI: RETURN
```

1. Czy smoki lubią poranne wstawanie?

Smok spał długo. Był to sen przyjemny, nikt nie próbował przeszkadzać dorodnemu smokowi. Przebudzenie nie jest rzeczą przyjemną i smoki wiedzą o tym równie dobrze, jak ludzie. Nasz smok leniwie garnął się do rzeczywistości — ciągle ziewał.

To „ziewanie" stacji dysków Timex-a jest spowodowane słabej jakości złączem krawędziowo-szufladowym i może być przyczyną wielu kłopotów. Złącze to po dłuższej przerwie w użytkowaniu stacji pokrywa się nalotem, który utrudnia przepływ prądu i pogarsza komunikację z systemem.

Można zauważyć to przy inicjacji systemu, lecz lepiej jest wpisać w trybie bezpośrednim (po restarcie przez **USR O**):

CLEAR 23856
CAT x

Jeśli katalog był krótki, należy powtórzyć katalogowanie aby zapełniony został cały obraz. Jeśli katalog jest prawidłowy, to oznacza, że smok brał już poranną kąpiel i nie jest senny.

Co zrobić, gdy katalog nie wyświetlił się prawidłowo? Są dwa wyjścia: proste — gorsze oraz trudniejsze — lepsze. Trudniejszym do wykonania, ale skuteczniejszym jest wymiana złącza na lepsze. W tym celu należy rozbierać interface, co powoduje, że jest droga dostępna tylko dla zaawansowanych majsterkowiczów.

Proste rozwiązanie jest dostępne dla każdego. W celu „udrożnienia" złącza trzeba przeczyścić styki szwankującego złącza. Najlepiej jest zrobić to przy pomocy preparatu w spray'u, typu „Kontaktol". Gdy nie jesteśmy szczęśliwymi posiadaczami spray'u, pozostaje nam zestaw zapałka+watka+benzynka.

Watkę nawiniętą na zapałkę ostrożnie zwilżamy benzyną i równie delikatnie przecieramy styki. Trzeba to robić uważnie, bo smok ogniem zieje!

Gdy cali i zdrowi (ewentualnie spaleni) podłączymy napęd, możemy ponownie sprawdzić poprawność transmisji. Powinna być bez zarzutu.

2. Co smoki jedzą na śniadanie?

Często zaniedbujemy smoki głodząc je przez nie podawanie im śniadania. Śniadanie dla smoka powinno być estetyczne i lekko strawne, zajmować tylko jedną jednostkę alokacji.

Tym śniadaniem jest oczywiście program START, który po restarcie systemu jest automatycznie uruchamiany.

Zamieszczony obok (lub poniżej/powyżej) program jest przydatny, gdy na dyskietce mamy kilka programów, których nie modyfikujemy, lecz używamy (gry, gry i gry).

W liniach **50—150** umieszczone są spacje jako dane dla DATA. Należy tam wpisać nazwy swoich programów tak, by długość ciągu znaków w nazwie była równa 8. Linia **530** wyłącza obsługę błędu TOS-a, zaś w **560**, **700** i **800** litery **AB** należy wpisać w trybie Ⓖ

Program nagrywamy na dyskietkę zleceniem

SAVE x„START"LINE 500

Jest to tylko przykład startu, który zainspiruje Was do pisania własnych — lepszych.

Smok to jednak zwierzę typowo domowe.

Maciej Pietraś

Ratowanie danych po zawieszeniu się komputera

Czasem w trakcie pracy nad programem w Turbo Pascalu zdarza się, że po kompilacji uruchomiony program, choć powinien wyświetlać np. wyniki obliczeń nie reaguje na ponaglające naciśnięcia klawiszy, choć przekroczył już przewidywany limit czasu. Oznacza to najprawdopodobniej zawieszenie się komputera.

Przyczyną takiego zachowania mogą być np. nieumiejętne odwołania do BIOS'u lub BDOS'u. Cóż, wypadek przy pracy. Gorzej jednak, jeśli w ferworze pracy przez dłuższy czas program nie był zapisywany na dyskietkę albo jego tekst znalazł się na ram-dysku — w takim wypadku czeka nas mozolne odtwarzanie z pamięci wprowadzonych do programu zmian.

Okazuje się jednak, że nie zawsze!!!

Jeżeli próba zresetowania komputera (SHIFT +EXTRA+EXIT) powiodła się, to w prosty sposób możemy odzyskać wprwadzone dane. Należy wgrać CP/M z dyskietki i wywołać program SID (3 strona dysków systemowych) i zleceniem W a: program. pas, adres-startowy, adres-końca zapisać na dyskietkę uratowany program. Oczywiście nazwę można dowolnie zmienić. Jako adres startowy podajemy odczytany wcześniej z menu TP początku tekstu programu (po napisie Text:) — warto go gdzieś zapisać! Adres końcowy powinien być dostatecznie wysoki aby objąć cały tekst, podanie A000 powinno wystarczyć.

Andrzej Wawrzyńczyk

PRZEDSZKOLAKI do komputerów!

```
 10 LET max=8
 20 LET s1=1+INT (max*RND)
 30 LET s2=s1-INT (s1*RND+0.5)
 40 CLS : BORDER 4: PRINT
 50 FOR i=1 TO max
 60 PRINT " ";i;" ";
 70 LET ile=i: GO SUB 500: PRINT ''
 80 NEXT i
 90 PRINT ''
100 LET ile=s2: GO SUB 500: PRINT "      ";
110 LET ile=s1-s2: GO SUB 500: PRINT ''
120 PRINT "-----> ";: FLASH 1
130 FOR i=1 TO 7: PRINT INVERSE i-2*INT (i/2);i;
140 NEXT i: FLASH 0
150 CIRCLE 210,130,35: CIRCLE 195,145,2
160 CIRCLE 225,145,2: GO SUB 400
170 IF INKEY$="" THEN GO TO 170
180 LET s=CODE INKEY$-CODE "0"
190 IF s<1 OR s>max THEN GO TO 220
200 PRINT AT 2*s-1,0; OVER 1; FLASH 1;"  "
210 FOR i=1 TO 500: NEXT i
220 OVER 1: GO SUB 400: OVER 0
230 IF s=s1 THEN GO TO 270
240 PLOT 195,115: DRAW 30,0,-1
250 FOR i=-15 TO -20 STEP -1: BEEP 0.75,i: NEXT i
260 GO TO 40
270 PLOT 195,120: DRAW 30,0,2
280 PRINT AT 0,1;: FOR i=1 TO 3: FOR j=0 TO 9
290 BEEP 0.025,10*i+j: PRINT j;
300 NEXT j: NEXT i
310 RUN
400 PLOT 195,120: DRAW 30,0,0.7: RETURN
500 FOR k=1 TO ile
510 PRINT " "; BRIGHT 1;" "; BRIGHT 0;
520 NEXT k: RETURN
```

Bartek niedługo kończy pięć lat. Z jego punktu widzenia Tata pracujący przy komputerze jest stałym elementem krajobrazu domowego, nic więc dziwnego, że on też chce „komputerować". Żeby mu to umożliwić, a przy okazji nauczyć go cyfr i dodawania (liczyć do dziesięciu już umie), napisałem dla niego program. Sukces był pełny — żeby przygotować coś do kolejnego numeru Bajtka muszę czekać do wieczora, bo włączanie komputera przy Bartku nieodmiennie kończy się tym, że to on zasiada przed klawiaturą.

Jak działa program? Jest bardzo prosty, i nie wymaga żadnej dodatkowej wiedzy. Na ekranie są wypisane cyfry od 1 do 9 (lub mniej — ich ilość jest określana przez zmienną max), obok nich znajdują się odpowiednie ilości jasnych kwadracików. Na dole jest tych kwadratów jeszcze kilka — trzeba je policzyć, i nacisnąć odpowiednią cyfrę na klawiaturze. Dobre odpowiedzi są nagradzane promiennym uśmiechem narysowanej w rogu ekranu twarzy, złe — „podkówką". Po dobrej odpowiedzi można grać dalej, po złej trzeba się poprawić, udzielając jeszcze raz odpowiedzi.

A więc — tatusiowie, starsi bracia i inni — do dzieła! Program jest krótki, można go wstukać w ciągu kilku minut, a potem — zawołać latorośl, i po trzech minutach dyskretnie wyjść z pokoju. Spokój do wieczora gwarantowany!

Marcin Borkowski



KILKA SŁÓW O KOMPRESJI TEKSTÓW

Wyobraź sobie, że postanowiłeś napisać grę — prawdziwą polską adventure w BASIC-u. Wymyśliłeś fantastyczny scenariusz, obmyśliłeś dokładnie konstrukcję programu, przez wiele dni siedzisz po 24 godziny na dobę i piszesz, piszesz, piszesz, aż tu nagle... klops! Skończyła się pamięć, którą dysponuje Twój komputer.

Co robić? W pierwszej chwili widzisz dwa rozwiązania — zrezygnować z pracy (teraz? tak blisko końca?) albo okroić scenariusz (taki dobry scenariusz!!!). Podpowiadam Ci trzecie rozwiązanie — przeczytaj ten artykuł!

Czy rzeczywiście musi zabraknąć miejsca w pamięci? Jedną z bardziej „pamięciożernych" instrukcji jest PRINT, bowiem jeżeli w swoim programie użyłeś jej w zwykły sposób, np. PRINT "Smok zieje ogniem!", to na zapamiętanie tekstu musiałeś zużyć tyle bajtów, z ilu znaków tekst się składał. Jeśli się dobrze zastanowisz, dojdziesz do wniosku, że musi istnieć sposób na lepsze od Twojego wykorzystanie pamięci (TO nie jest żadna aluzja na temat sklerozy). Wystarczy przypomnieć sobie Hobbita lub Sherlocka, w których pojawiający się na ekranie tekst, a dokładniej ilość znaków, znacznie przekracza pojemność RAM-u. Sugeruje to, że teksty pojawiające się na ekranie nie są zapamiętane według zasady jeden bajt = jedna litera (lub inny znak pisarski, np. cyfra lub znak przestankowy).

Ponieważ problem braku pamięci jest tak stary jak komputery (jeżeli brak pamięci rozumieć odpowiednio szeroko, to problem jest tak stary jak ludzkość), nad zmniejszaniem ilości pamięci potrzebnej do zapamiętania tekstu pracowano już nieraz. Pewne informacje na ten temat można znaleźć w książkach o bazach danych, gdzie problem ten często występuje. Niestety, prezentowane tam metody wymagają zastosowania kodu maszynowego. Program napisany w BASIC-u byłby bardzo wolny, gdyż musiałby operować na pojedynczych bitach mozolnie wyliczanych ze znanych wartości bajtów. Nie tędy więc droga.

Zastanówmy się nad tym, dlaczego tracimy pamięć — może to pozwoli na znalezienie sposobu jej zaoszczędzenia. Do zapisywania komunikatów drukowanych przez program używa się 32 liter — małych i wielkich, 10 cyfr, około 6 znaków przestankowych (;,.?! — — z pozostałych można zrezygnować) i spacji, czyli 81 znaków kodu ASCII. Ponieważ bajt może przyjmować 256 różnych wartości, pozostaje 175 wartości nie wykorzystanych. I tu jest właśnie pies pogrzebany, bo korzystamy tylko z 32% możliwości oferowanych przez pamięć (mowa tu oczywiście o wykorzystaniu pamięci komputera, a nie naszych bliźnich). Co robić, żeby poprawić ten wynik? Może wykorzystać pozostałe 175 liczb do kodowania końcówek — np. „isz", która występuje dość często, w słowach „stoisz", „widzisz", może zapamiętać w ten sposób „rz", „sz", „ch"? Niestety, w ten sposób uda się wykorzystać najwyżej kilkanaście spośród 175 możliwości, czyli ciągle większość wartości przyjmowanych przez pojedynczy bajt pozostanie nie wykorzystana. W takim razie może podejść do sprawy bardziej systematycznie? Gdyby tak zbadać częstotliwość występowania w języku polskim zestawów dwu i trójliterowych, i wziąć pod uwagę te występujące najczęściej?

Tą właśnie drogą doszedłem do zasadniczej idei, na której opiera się moja propozycja. W ostatecznej wersji wprowadziłem jeszcze jedną zmianę — wyrzuciłem wszystkie wielkie litery, wprowadzając następującą zasadę — jeżeli poprzedni bajt ma wartość zero, pierwsza litera następnego zestawu znaków mą być wydrukowana jako wielka litera.

Oczywiście żeby móc wybrać odpowiednie pary i trójki liter, a także określić ich ilości pośród możliwych 255 znaków (zero już zostało wykorzystane), musiałem zbadać statystycznie język polski. Użyłem do tego celu tekstu składającego się z fragmentów kilku znanych książek, które miałem pod ręką. Na podstawie tych wyników wybrałem odpowiednie zestawy znaków, które są w programie umieszczone w tablicach **a$**, **b$** i **c$** (odpowiednio znaki jedno dwu i trzyliterowe). (Jeśli ktoś myśli, że ich uzyskanie było łatwe, niech weźmie pod uwagę trzy liczby — 35 kB dostępne RAM na Spectrum, tekst zajmował około 15 kB, różnych zestawów znaków trzyliterowych jest — bagatelka — około 36 tysięcy, nie mówiąc już o takim drobiazgu jak program, który całą pracę miał wykonać!). Nie twierdzę, że nie ma lepszego zestawu, dającego większe oszczędności. Przy pomocy proponowanego, na zapamiętanie tekstu składającego się ze stu znaków zużywa się średnio 55–59 bajtów, czyli na jedną literę przypada mniej więcej 4,6 bitu.

Żeby móc skorzystać z zaproponowanej metody, trzeba jeszcze znaleźć sposób kodowania tekstu. Wybrałem metodę nie dającą wprawdzie optymalnych wyników, ale łatwą w realizacji — i co ważne — działającą. Umówmy się (dla wygody), że od tej pory znakiem będziemy nazywać nie tylko pojedynczą literę, ale również zbiór dwu lub trzech liter. Weźmy pod uwagę słowo „kodowanie". Najpierw oddzielamy od niego trzy pierwsze litery — „kod", i szukamy pośród znaków trzyliterowych. Nie znajdziemy go, bo takiego znaku nie ma. Skracamy więc znak o jedną literę — zostaje „ko", i szukamy wśród dwuliterowych — jest taki, ma numer 80. Zapamiętujemy jako pierwszy bajt 80, odejmujemy od słowa „kodowanie" dwie pierwsze litery, i powtarzamy całą operację z tym co zostało, czyli ze słowem (?) „dowanie". Gdybyśmy nie znaleźli znaku „ko" wśród dwuliterowych, zaczęlibyśmy szukać między pojedynczymi literami litery „k". Ostatecznie, po przeprowadzeniu operacji do końca, słowo „kodowanie" powinno zostać zapamiętane w czterech bajtach — ko-do-wa-nie (80, 97, 85, 241). W praktyce algorytm jest trochę zmieniony, żeby dawał sobie radę z dużymi literami, ale nie są to zmiany istotne.

Odszyfrowanie jest już bardzo proste — nie opisuję metody, bo można ją łatwo zrozumieć analizując program dekodujący, zresztą jest oczywista.

Pozostają jeszcze cztery uwagi. Pierwsza dotyczy polskich znaków diakrytycznych, czyli liter ą, ę itd. Jeżeli do ich przedstawienia na ekranie używa się UDG, nie mogą występować jako małe i wielkie — przynajmniej przy zastosowanych przeze mnie założeniach, że kody ASCII dużej i małej litery zawsze różnią się o tyle samo (konkretnie 32). Żeby obejść ten problem najlepiej zaprojektować własny zestaw znaków, wykorzystując do swoich celów litery v, q, a także znaki [,], {, } /. Ten właśnie sposób zastosowałem pisząc „Puszkę Pandory". Gra ta jest wprawdzie napisana w kodzie maszynowym, ale korzysta z opisanego sposobu kompresji tekstu.

Druga uwaga dotyczy tego, kiedy warto zastosować moją metodę. Zapamiętanie przy jej pomocy tekstu mającego sto znaków, zamiast oczekiwanego zysku przyniesie straty pamięci. Jest to oczywiste, jeżeli uświadomimy sobie, że sam program dekodujący i tablice, w których zapamiętane są znaki zajmują około 600 bajtów.

Średnio na zapamiętanie tekstu składającego się ze stu znaków zużywamy około 57 bajtów. Stosowanie metody ma sens dopiero wtedy , kiedy łączna długość programu, danych i zakodowanego tekstu jest mniejsza niż długość samego tekstu bez kodowania. Rozwiązanie prostej nierówności prowadzi do wniosku, że tekst musi mieć co najmniej 1400 znaków.

Trzecia uwaga dotyczy pewnej niezgodności tego, co robi program, z tym co napisałem. Kto chce się przekonać, że tak jest, niech spróbuje zakodować przy pomocy programu słowo „kodowanie", które podałem jako przykład. Nie zmieści się ono w czterech bajtach, tylko w pięciu. Wiąże się to z tym, że znaki przestankowe występują (oprócz myślnika) prawie zawsze przed spacją, toteż umieściłem je w tablicy **b$**, razem ze znakami dwuliterowymi. Nie musimy jednak pamiętać o dopisywaniu spacji na końcu każdego tekstu — program robi to sam, a w efekcie kodowany tekst będzie o tą jedną spację dłuższy — „kodowanie ".

Ostatnia sprawa, na którą należy zwrócić uwagę, to konieczność utworzenia specjalnej tablicy, w której będą zapamiętane adresy początkowe i końcowe kolejnych tekstów, ale to już problem do rozwiązania dla tych, którzy będą korzystać z mojej metody. Użytkowników Spectrum przestrzegam przed zapamiętywaniem tych danych w tablicy liczbowej — oznacza to stratę sześciu bajtów na każdym zapamiętanym tekście. Nie podaję dalszych komentarzy — jeśli ktoś chce skorzystać, niech sam też trochę pomyśli.

Na koniec krótka instrukcja obsługi zamieszczonego programu. Składa się on z trzech części. Pierwsza z nich (linie 1–125) służy do przygotowania i ewentualnego nagrania na taśmę danych — tablic **a$**, **b$** i **c$** oraz liter ą, ę itd. — UDG od A do H odpowiadają kolejno literom ą, ć, ę, ł, ń, ó, ś i ż, które służy jako „żet" i „ziet". (UDG na listingu występuje jako duże, podkreślone litery). W liniach od 200 do 820 znajduje się program kodujący wprowadzony tekst. Po niewielkich przeróbkach można go użyć do przygotowania w pamięci serii zakodowanych tekstów, trzeba tylko przy tym wiedzieć, że zmienna **r** oznacza adres początku zakodowanego tekstu, a zmienna **aktadr** (po zakończeniu kodowania) zwiększony o 1 adres końca kodu. Trzecia część programu umieszczona w liniach 1000–2000 to dekoder. Do jego działania potrzebne są tablice a$, b$, c$ i wartości zmiennych **r** i **aktadr**. Program był wprawdzie napisany na Spectrum, ale powinien dać się łatwo przenieść na inne typy komputerów. Jedyny problem może wystąpić w liniach 600 i 920. W linii 600 zmienna **k$** staje się łańcuchem składającym się z **dz** pierwszych znaków zmiennej **w$**, w linii 900 zmienna **w$** zostaje skrócona o **dz** pierwszych znaków. W linii 205 program odwołuje się do zmiennej systemowej RAMTOP, żeby sprawdzić gdzie w pamięci znajduje się miejsce na zapamiętywanie zakodowanego tekstu — wykonanie w linii 5 instrukcji CLEAR powoduje oprócz zmiany RAMTOP-u również skasowanie zmiennej **r**, w której zapamiętany był RAMTOP wprowadzony przez użytkownika.

Nie napisałem nic na temat jeszcze jednego zastosowania opracowanego przeze mnie algorytmu — można go z powodzeniem stosować do szyfrowania tekstów, które chcemy uchronić przed niepowołanymi Czytelnikami. Nic dodać, nic ująć — dla każdego, kto zrozumiał założenia, na których opiera się moja metoda jasne jest, ile mamy możliwości wybrania innego kompletu zestawów dwu i trzyliterowych. Nie wyobrażam sobie żadnego algorytmu, który mógłby sobie dać radę z tak zaszyfrowanym tekstem — a o to przecież chodzi.

Marcin Borkowski

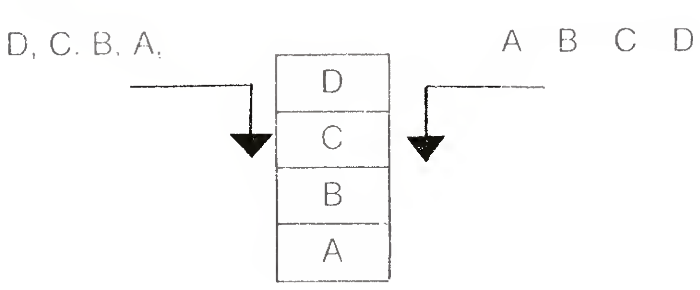
```
   5 INPUT "Ramtop? ";r: CLEAR r
  10 FOR i=USR "a" TO USR "i"-1: READ a: POKE i,a: NEXT i
  15 DATA 0,0,56,4,60,68,60,8,0,8,28,32,32,32,28,0
  20 DATA 0,0,56,68,120,64,60,8,0,16,20,24,16,16,12,0
  25 DATA 0,16,120,68,68,68,68,0,0,16,56,68,68,68,56,0
  30 DATA 0,16,56,64,56,4,120,0,0,16,124,8,16,32,124,0
  35 DATA "↑"," ","a","b","c","d","e","f","g","h","i","j","k
","l","m","n","o","p","r","s","t","u","w","y","z","A","B","C
","D","E","F","G","H","0","1","2","3","4","5","6","7","8","9
","-"
  40 DATA "nie","rze","dzi","siC","prz","wie","rzy","ego","c
zy","isz","sz ","ch ","ej ","na ","   "
  45 DATA "e ","ie","a ","ni","o "," p","ym"," s"," n"," w",
"i "," z","rz","na"," t"," o","po","ch","cz"," d","ze","zy",
"wi"," j"," m","st","aD","C ","ia","iC","mi","pr","m ","za",
"ow","si","sz","ko"
  50 DATA "ro","D "," b","dz","wa","ra","ta"," i","Da"," c",
" k","ak","k ","te","to","z ","do","od","w ","zl","n ","wy",
"on","an","je","le","ci","j ","B ","al","am","eg","em","ej",
"at","by","ne","ty"
  55 DATA "li","ka","ki","He"," a","ed","go","ob","or","yc",
"Do","A "," r"," H","ma","ad","da","en","os","Dy","ac","ja",
"yD","ar","er","ok","as","u ","ny"," g","aj","aw","no","ot",
"ec","tr","zn","ym"
  60 DATA " u","la","es","kr","dy","ek","ic","mo","om","pa",
"sk","wo","H "," l","in","jA","pi","bi","Ac","Fw","my","d ",
"ez","iD","t ","ol","ys","zo","c ","re","tw","we","ce","el",
"kt","ku","sp","ak"
  65 DATA "mu","ud","zc","oH","Fr","Hy","aB","dn","im","nA",
"yn","no","oc","oz","ry","ws","cj","gD","ju","so","tF","ur",
" h","et","ik","nn","sa","us","uH","Gc","br","cy","dw","ew",
"il","is","lu","oG","yl","Gl","? ","! ",", ",". "
  70 DIM a$(44): DIM b$(197,2): DIM c$(15,3)
  75 FOR i=1 TO 44: READ a$(i): NEXT i
  80 FOR i=1 TO 15: READ c$(i): NEXT i
  85 FOR i=1 TO 197: READ b$(i): NEXT i
  90 INPUT "Save danych? t/n "; LINE o$
  95 IF o$="n" OR o$="N" THEN GO TO 200
 100 INPUT "Save nazwa? "; LINE o$
 105 SAVE o$ DATA a$(): SAVE o$ DATA b$()
 110 SAVE o$ DATA c$(): SAVE o$CODE USR "a",64
 115 PRINT #1;"Przewiń taśmę do weryfikacji."
 120 VERIFY o$ DATA a$(): VERIFY o$ DATA b$()
 125 VERIFY o$ DATA c$(): VERIFY o$CODE USR "a",64
 200 CLS: INPUT "Tekst do kodowania? ";AT 3,0; LINE w$
 201 PRINT "Długość tekstu ";LEN w$''
 203 LET w$=w$+" "
 205 LET r=256*PEEK 23731+PEEK 23730+1
 210 LET q=1: LET aktadr=r: LET k$=""
 220 GO SUB 500
 230 PRINT ''"ilość znaków ";aktadr-r
 300 PRINT '"Początek kodu ";r'"Koniec kodu    ";aktadr-1''
 310 GO SUB 1000: GO TO 200

 500 REM ***************************
 501 REM **  Procedura kodująca  **
 502 REM ***************************
 505 LET dz=3: PRINT BRIGHT q;k$;: LET q=1-q
 510 IF LEN w$>=dz THEN GO TO 600
 520 LET dz=dz-1
 530 IF dz>0 THEN GO TO 510
 540 IF w$="" THEN RETURN
 600 LET k$=w$( TO dz): GO TO 650+50*dz
 700 FOR i=1 TO 44
 710 IF a$(i)=k$ THEN LET kod=i: GO TO 900
 720 NEXT i
 730 IF CODE w$>64 AND CODE w$<91 THEN GO TO 733
 731 PRINT "Nie znam znaku ";k$: STOP
 733 LET w$(1)=CHR$ (CODE w$+32)
 736 LET w$="↑"+w$: LET dz=3: GO TO 510
 750 FOR i=1 TO 197
 760 IF b$(i)=k$ THEN LET kod=44+i: GO TO 900
 770 NEXT i: GO TO 520
 800 FOR i=1 TO 15
 810 IF c$(i)=k$ THEN LET kod=241+i: GO TO 900
 820 NEXT i: GO TO 520
 900 POKE aktadr,kod-1: LET aktadr=aktadr+1
 920 LET w$=w$(dz+1 TO): GO TO 500
1000 REM *****************************
1001 REM **  Procedura dekodująca  **
1002 REM *****************************
1005 LET duza=0: FOR i=r to aktadr-1
1010 LET p=1+PEEK i
1020 IF p=1 THEN LET duza=1: NEXT i
1030 IF p<45 THEN LET d$=a$(p): GO TO 1060
1040 IF p<242 THEN LET d$=b$(p-44): GO TO 1060
1050 LET d$=c$(p-241)
1060 IF duza THEN LET d$(1)=CHR$ (CODE d$-32): LET duza=0
1070 PRINT d$;: NEXT i
1100 INPUT "Nagrywanie kodu? t/n "; LINE o$
1105 IF o$="n" or o$="N" THEN STOP
1110 INPUT "Nazwa bloku kodu? "; LINE o$
1130 SAVE o$CODE r,aktadr-r
1140 PRINT #1;"Przewiń taśmę - weryfikacja": VERIFY o$CODE
2000 RETURN
```

STRUKTURY DANYCH VI

Naszym tematem będzie dziś praktyczne zastosowanie stosu. Nie oznacza to powrotu do zwyczaju polowania na czarownice, lecz jedynie powrót do struktury danych, którą opisaliśmy miesiąc temu. Nie udało się jednak zmieścić opisu rozsądnych zastosowań tej struktury i właśnie takimi zastosowaniami zajmiemy się teraz.

Przypomnijmy tylko krótko, że stos daje dostęp do danych zgodnie z zasadą: ostatni przybył pierwszy wychodzi. Można to zilustrować następującym szkicem:



JAK OBSŁUGIWAĆ PRZERWANIA

Zapomnijmy teraz na chwilę o stosie i przyjrzyjmy się pracy procesora centralnego w komputerze. Wykonuje on kolejno rozkazy zapisane w pamięci operacyjnej. Przez cały czas procesor pamięta w specjalnym rejestrze, tzw. liczniku rozkazów (LR) adres następnego rozkazu do wykonania (oczywiście wykonanie każdego rozkazu zmienia zawartość LR). Jeśli zdarzy się coś co wymaga natychmiastowej reakcji procesora (np. użytkownik nacisnął klawisz funkcyjny) to ma miejsce tzw. przerwanie i zamiast wykonać następny rozkaz w/g licznika trzeba zacząć wykonywac podprogram obsługujący przerwanie. Maszyna robi to wstawiając do licznika rozkazów adres początku podprogramu obsługi*). Ale po zakończeniu obsługi przerwania musimy kontynuować poprzedni program od miejsca w którym został przerwany. Aby to zrobić musimy w momencie przerwania zapamiętać wartość licznika rozkazów. Umówmy się, że np. w komórce pamięci o adresie 100. Po prostu obsługa przerwania zaczyna się od zapisania zawartości LR pod adres 100. Wykonanie osbługi przerwania może wymagać sporo czasu i w tym czasie nasz użytkownik może znowu coś nacisnąć, co wygeneruje następne przerwanie i spowoduje ponowne wywołanie podprogramu jego obsługi. Pierwszą czynnością tego podprogramu jest znowu zapisanie ostatniej zawartości LR pod adres 100. I oczywiście w tym momencie zamazany został adres powrotu do programu z poprzedniego przerwania.

Prawidłowo powinno być tak: przerwanie 2 jest obsłużone do końca, z komórki 100 pobieramy adres powrotu, który wskazuje na miejsce (adres rozkazu w pamięci) w którym przerwano obsługę przerwania 1, przerwanie 1 jest obsłużone do końca, z komórki 100 pobieramy adres powrotu do programu i kontynuujemy wykonanie programu. Tak się niestety zrobić nie da — z tego samego miejsca nie można pobrać kolejno kilku różnych wartości. Chyba, że zamiast zwykłej komórki użyjemy stosu. Zawsze gdy przychodzi przerwanie, aktualną zawartość LR pakujemy na stos. To, że poznane adresy będziemy zdejmować w odwrotnej kolejności idealnie nam pasuje. Przecież jeśli pojawiły się kolejno (każde następne przed zakończeniem poprzedniego) przerwania P1, P2, P3, P4, P5, to gdy wreszcie udało się pomyślnie i bez przerwy zakończyć P5, należy wrócić tam skąd P5 było wywołane, a więc do P4, po jego zakończeniu do P3 itd. narysujmy ten przykład aby mieć pewność, że wszystko jasne:

Po uruchomieniu obsługi przerwania P1 mamy na stosie:

adres powrotu do programu

gdy w czasie wkonywania obsługi P1 wystąpiło przerwanie P2:

adres powrotu do obsługi P1
adres powrotu do programu

gdy w czasie obsługi P2 nastąpiło przerwanie P3:

adres powrotu do obsługi P2
adres powrotu do obsługi P1
adres powrotu do programu

itd., aż wreszcie w czasie obsługi P5 mamy na stosie:

adres powrotu do obsługi P4
adres powrotu do bsługi P3
adres powrotu do obsługi P2
adres powrotu do obsługi P1
adres powrotu do programu

Kończąc obsługę P5 zdejmujemy ze stosu adres P4 i kontynuujemy od momentu przerwania. Stos wygląda następująco:

adres powrotu do obsługi P3
adres powrotu do obsługi P2
adres powrotu do obsługi P1
adres powrotu do programu

Dalej działamy analogicznie, choć oczywiście zanim wrócimy do poziomu programu mogą występować następne przerwania i stos będzie rósł zamiast maleć. Niemniej jednak nigdy nie zgubimy ścieżki powrotu. A jeśli ktoś nie jest przekonany, że warto używać tu stosu to niech spróbuje rozwiązać ten problem w inny sposób, i porównać rozwiązania. Proszę tylko pamiętać, że w czasie wykonywania podprogramu obsługi konkretnego przerwania może znowu wystąpić dokładnie to samo przerwanie i ten sam podprogram będzie uruchomiony po raz kolejny przed zakończeniem poprzedniego wykonania.

TŁUMACZENIE WYRAŻEŃ ARYTMETYCZNYCH

Procesor komputera może wykonywać operacje arytmetyczne po jednej, w takiej kolejności w jakiej są podawane przez sterowanie programu. My, ludzie, jesteśmy przyzwyczajeni do zapisywania wyrażeń zgodnie z pewnymi regułami, dzięki którym interpretacja wyrażenia jest zupełnie jednoznaczna. Np. wiemy że aby obliczyć wartość:

a + b × c

trzeba pomnożyć b × c i dopiero wtedy dodać wynik do a, natomiast w przypadku:

a × (b + c) + 1

najpierw dodajemy b + c, następnie a mnożymy przez wynik [illegible] i wreszcie otrzymaną wartość mnożymy [illegible]

Wykonując [illegible] musimy pamiętać o tym, że mnożenie i dzielenie wykonujemy przed dodawaniem i odejmowaniem (mówimy, że mnożenie i dzielenie mają wyższy priorytet niż dodawanie i odejmowanie, zaś mnożenie i dzielenie mają taki sam priorytet). Dalej, działania w nawiasach wykonujemy przed pozostałymi działaniami, i wreszcie pamiętajmy, że operacje wykonujemy od lewej do prawej **). Zwykle ustalenie dobrej kolejności działań nie sprawia nam trudności dzięki wprawie nabytej podczas ćwiczeń w szkole podstawowej. Jednak obliczenie np. wartości wyrażenia:

11+(10×(12+34−34/(21−12)−88/7) 11×(5+90)+7)/10−6

choć nie jest wysiłkiem na miarę nagrody Nobla, to wymaga chwili koncentracji.

Ale obliczanie wartości takich potworków nie jest naszym celem. Jako informatycy chcemy żeby robił to za nas komputer. I robi. W językach programowania wyższego rzędu można po prostu pisać dowolne wyrażenia arytmetyczne prawie tak jak nauczyliśmy się tego w szkole, natomiast zadaniem translatora jest zapisanie tego wyrażenia w postaci elementarnych operacji (dodawanie, mnożenie itd.), w takiej kolejności w jakiej komputer ma je wykonywać, czyli w kolejności wynikającej z nawiasów i z priorytetów operacji arytmetycznych.

Jeśli ktoś chce trochę poćwiczyć umysł to może teraz przerwać lekturę i spróbować zaprojektować program translacji wyrażeń arytmetycznych na ciągi rozkazów w języku maszynowym. Ja natomiast opiszę rozwiązanie znane od wielu lat ***), w którym do tłumaczenia wyrażeń używany jest stos.

Ponieważ chodzi nam przede wszystkim o przedstawienie koncepcji algorytmu a nie żmudną analizę szczegółów technicznych ograniczymy się do translacji bardzo prostych wyrażeń, zawierających jedynie nazwy zmiennych oraz operatory czterech podstawowych działań: +, −, *, /. Operatorom tym nadamy priorytety: dla +, − priorytet jest równy 1, dla *, / priorytet jest równy 2.

Reguły translacji są niezwykle proste. Zaczynamy mając pusty stos i wczytujemy kolejne elementy wyrażenia poczynając od lewej strony. Jeśli elementem jest zmienna to przepisujemy ją bezpośrednio na wyjście, jeśli operator to badamy kilka możliwości:

1. Gdy stos pusty to wpisujemy operator na stos;
2. Gdy na wierzchołku stosu znajduje się operator o priorytecie mniejszym niż nasz badany, to dopisujemy nasz badany na stos;
3. Jeśli na wierzchołku stosu znajduje się operator o priorytecie większym lub równym od naszego, to zdejmujemy go ze stosu i wysyłamy na wyjście. Natomiast dla wczytanego powtarzamy całe badanie od punktu 1.

Proces kończymy po wczytaniu całego wyrażenia. Ostatnią czynnością jest wypisanie na wyjście całej zawartości stosu. Pamiętamy z opisu stosu, że jedyna możliwa metoda wykonania tej operacji to pobieranie po kolei wierzchołka stosu i przesyłanie pobranej wartości na wyjście.

To co otrzymamy nie jest jeszcze ostateczną postacią lecz stadium pośrednim, jednak na tyle ciekawym i ważnym, że poświęcimy mu trochę uwagi. Na razie jednak pragnę podkreślić chyba oczywisty wniosek, że zaprogramowanie powyższego algorytmu jest dziecinnie łatwe. Natomiast zamiast uzasadnienia poprawności, które nie zmieści się w tym artykule popatrzmy na przykład działania algorytmu dla wyrażenia:

a+b*c a/b

wejście	stos	wyjście	wyrażenie wynikowe
a	pusty	a	a
+	+		a
b	+	b	a.b
*	* +		a.b
c	* +	c	a.b.c
−	+ pusty −	* +	a.b,c * a.b.c,*.+ a.b.c,*.+
a	−	a	a.b.c,*.+.a
/	/ −		a.b.c * +.a
b		b	a.b.c * +.a.b
	pusty	/	a.b.c.*.+.a.b./ a.b.c * +.a.b./,

Otrzymaliśmy następujący wynik: a.b.c*.+.a.b./.–. Jest to zapis naszego wyrażenia arytmetycznego w innej postaci — operatory znajdują się nie między argumentami lecz po nich. w tej konwencji wyrażenie a+b zostało by zapisane jako a.b,+. Zapis a,b,c.* + oznacza, że należy wykonać kolejno operacje: b*c, a następnie dodać otrzymany wynik do a. Tak więc a,b,c.* + odpowiada tradycyjnemu zapisowi a+b*c. Omówiony sposób zapisywania operacji arytmetycznych nosi nazwę Odwrotnej Notacji Polskiej (ONP).

Aby na podstawie ONP wygenerowac ostateczny kod maszynowy obliczający wartość tego wyrażenia posłużymy się znowu stosem. Algorytm drugiej fazy translacji jest następujący:

Zaczynamy od pustego stosu. z wejścia pobieramy kolejne elementy wyrażenia w ONP. poczynając od lewej. Jeśli wczytaliśmy zmienną to wpisujemy ją po prostu na stos. Jeśli jest to operator to pobieramy ze stosu dwa ostatnie elementy i wykonujemy na nich działanie wskazane tym operatorem a wynik umieszczamy na stosie.

Proces kończymy po przetworzeniu całego wyrażenia. Na stosie powinien wtedy zostać jeden element — obliczona wartość wyrażenia.

Oczywiście powyższy algorytm jest tylko szkicem mającym pokazac jak przejść od zapisu w ONP do zapisu ciągu operacji arytmetycznych w takiej kolejności w jakiej powinien je wykonywać procesor. Aby wygenerować kod wynikowy musimy wstawić odpowiednie czynności w miejscu gdzie napisaliśmy krotko: „wpisujemy zmienną na stos", „pobieramy ze stosu dwa ostatnie elementy i wykonujemy na nich działanie wskazane tym operatorem a wynik umieszczamy na stosie". Te czynności to zapisywanie instrukcji maszynowych ładujących wartości zmiennych (czyli odpowiednich komórek pamięci) do rejestrów procesora, instrukcji wykonujących wskazane operatorem działanie arytmetyczne, instrukcji zapisujących wynik w odpowiedniej komórce pamięci. Jest to sporo pracy, jednak pozostaje ona poza naszymi dzisiejszymi zainteresowaniami, gdyż operacje te trzeba wykonywać niezależnie od metody jakiej użyliśmy aby przejść od „ludzkiej" postaci wyrażenia do operacji arytmetycznych ustawionych w kolejności w jakiej ma je wykonywac procesor. Widziana w taki sposób druga faza translacji wyrażenia jest również prosta.

Mam nadzieję, że powyższe przykłady, w których zastosowanie stosu pozwoliło skonstruować proste algorytmy rozwiązania trudnych (a równocześnie mających ogromne znaczenie praktyczne) problemów, przekonały Was o przydatności tej, na pozór dziwacznej i nieprzydatnej struktury.

*) Proces obsługi przerwania jest zwykle dużo bardziej skomplikowany i wymaga przechowywania oprocz LR także zawartości innych rejestrów.
**) Arytmetyka uczy, że dodawanie i mnożenie są operacjami przemiennymi, czyli
a*b = b*a
a więc kolejność zapisania operandów nie powinna mieć wpływu na wynik. Jednak w programach, ze względu na możliwość wystąpienia tzw. ubocznych efektów procedur kolejność ta ma znaczenie nawet dla operacji przemiennych, stąd istotne jest określenie kolejności interpretacji wyrażeń, zwykle właśnie od lewej do prawej.
***) Bardziej szczegółowy opis tego zadania i jego rozwiązania można znaleźć np. w książce prof. Turskiego, „Propedeutyka informatyki", PWN Warszawa 1975 r.

Andrzej Krul

ROZWIĄZANIE PROBLEMU KOLEJKI

W poprzednim odcinku zostawiliśmy bez odpowiedzi pytanie: jak optymalnie zrealizować kolejkę w oparciu o listę jednokierunkową. Sposób jest bardzo prosty: otóż należy dopisywać nowe elementy na końcu, zaś wychodzące z kolejki pobierać z początku listy. Aby to sprawnie wykonywać, musimy zapamiętać koniec listy i za każdym razem dopisując nowy element przesuwać tę strzałkę tak, aby zawsze wskazywała koniec. Dzięki temu zawsze wiemy, gdzie dopisać nowy element bez żadnych operacji przeszukiwania. Także pobieranie nie wymaga przeszukiwania, gdyż początek listy mamy standardowo zapamiętany i potrafimy z niego łatwo usuwać elementy.

Jeśli ktoś ma problemy z wyobrażeniem sobie działania tak zorganizowanej kolejki, to proponuję, aby wykonał kilka rysunków takich, jak robiliśmy w zeszłym miesiącu, i na ich podstawie przekonał się o prawidłowości rozwiązania.

Ceny mogą być wyższe ze względu na inflację, a poza tym obowiązuje tzw. „przelicznik dolarowy"

	Giełda Bajtka	Sklep Bajtka	Komis	Pewex	RFN	Baltona	CSH i inne
	tys. zł	tys. zł	tys. zł	$	DM	$	tys. zł
SINCLAIR							
ZX 81	120	120	—	—	—	—	—
ZX Spectrum 48	400	350	550	—	80	—	—
ZX Spectrum +	450	550	700	—	90	—	—
Timex 2048	400	400	400	—	—	—	240
ZX Spectrum 128+	550	—	—	—	—	—	—
ZX Spectrum 128+2	—	—	—	—	140	—	—
ZX Spectrum 128+3	—	—	—	—	280	—	—
drukarka Seikosha GP 50s	—	280	—	—	—	—	—
Interface Kempston	25	20	—	—	10	—	9.3
COMMODORE							
Commodore 64	900	650	710	199	290	155	950
VC 20	200	180	—	—	—	—	—
C 16	300	250	—	—	80	—	—
C 116	350	240	—	—	70	—	—
C Plus 4	450	450	380	—	150	—	—
C 128	1100	1100	—	—	399	—	—
C 128 D	1900	—	—	450	820	—	—
Amiga 500	2300	2900	—	—	899	—	—
Magnetofon 1531	150	170	170	48	30	225	—
Stacja dysków Oceanic	700	700	—	—	320	170	—
Stacja dysków 1571	1000	1000	—	199	460	—	—
Drukarka LCIOC	1300	1300	—	—	260	230	—
ATARI							
Atari 800 XL	600	700	470	—	160	—	—
Atari 65 XE	700	650	520	127	—	—	—
Atari 130 XE	800	850	—	199	220	—	—
Atari 520 ST	2000	2000	—	—	—	—	—
Atari 1040 ST	—	—	—	—	1140	—	—
Magnetofon XC 12	140	140	—	36	40	—	—
Stacja dysków 1050	900	900	750	185	300	—	—
Stacja dysków 520 STM	—	700	—	—	—	—	—
Drukarka 1029	400	650	—	—	—	—	—
AMSTRAD							
Amstrad 464 mono.	900	900	—	—	350	—	—
Amstrad 664 mono.	—	1200	—	—	—	—	—
Amstrad 6128 mono.	—	—	—	—	670	—	—
Amstrad PCW 8256	—	—	—	—	—	—	—
Amstrad PCW 8512	—	—	—	—	—	—	—
Amstrad PCW 9512	—	—	—	—	—	—	—
Stacja dysków do 464	—	—	—	—	380	—	—
SHARP							
Sharp MZ 700	—	360	—	—	—	—	—
Sharp MZ 800	—	420	—	—	—	—	—
Dyskietki 5.25 cala	2	2-2.5	3	1	0.7	11	4-9
Dyskietki 3.5 cala	7.5	7.5-9	7-9	—	5	2.5	9-10
Dyskietki 3 cale	11	—	—	—	6	3	10
Joystick	21	16-26	20-30	5	10	10	14
Monitor Neptun	60	80	—	—	—	—	68

INDYWIDUALNY BANK DANYCH

Łukasz Atkonis, uczeń lat 14. Posiada mikrokomputer ATARI 65 XE z magnetofonem XC-12 oraz joystick. Oprogramowanie: około 300 gier i programów użytkowych. Zainteresowania: informatyka, muzyka, pisanie programów użytkowych. Proponuje wymianę programów oraz gier. Adres: 05-807 Podkowa Leśna, ul. Dzierżyńskiego 27.

Lech Mazur, uczeń lat 13. Posiada Commodore 64 ze stacją dysków 1541. Interesuje się informatyką i sportem. Proponuje wymianę oprogramowania. Adres: 34-400 Nowy Targ, ul. Królowej Jadwigi 8.

Michał Idzik, lat 11. Posiada ATARI 800 XL, magnetofon ATARI 1010. Oprogramowanie: głównie gry. Zainteresowania: fantastyka, numizmatyka, komputery. Proponuje wymianę oprogramowania. Adres: 41-208 Sosnowiec, ul. Kalinowa 83/8.

Rafał Jankowski, lat 21. Prosi posiadaczy AMIGI 500 o kontakt w celu zdobycia oprogramowania oraz literatury. Interesuje się informatyką, motoryzacją i fotografią. Adres: 34-400 Nowy Targ, Pl. Słowackiego 17.

Błażej Begger, uczeń LO. Posiada mikrokomputer Schneider CPC 464 pragnie nawiązać kontakt z posiadaczami tego komputera w celu wymiany gier i programów. 89-600 Chojnice, ul. Gdańska 57.

Krzysztof Firek, uczeń, 17 lat. Posiada COMMODORE 64, magnetofon, monitor, final II oraz około cztery tysiące programów. Proponuje wymianę programów, czasopism i doświadczeń. 32-400 Myślenice, ul. Sobieskiego 32 c.

Beata Koch, uczennica VII klasy — lat 14. Posiada TIMEX-a 2048. Oprogramowanie: około 400 gier i programów użytkowych. Proponuje wymianę programów. Adres: 61-673 Poznań, Oś. Kraju Rad 7/72.

Olaf Klisiński, lat 13. Posiada mikrokomputer ZX Spectrum +, magnetofon MK 450. Oprogramowanie: 250 programów głównie gier. Proponuje wymianę oprogramowania oraz gier. Adres:40-883 Katowice, ul. B. Chrobrego 17/130.

Grzegorz Piekarz, lat 29 — nauczyciel. Zaprasza do wymiany programów na Schneidera CPC 6128. Adres: 85-614 Bydgoszcz, ul. Głowackiego 25/11.

Łukasz Ścieżar, uczeń lat 12. Posiada mikrokomputer TIMEX 2048. Oprogramowanie: około 50 gier. Proponuje wymianę gier. Adres: 32-050 Skawina, ul. Słoneczna 1 A m 54.

WSZYSTKO DLA WSZYSTKICH

AKCES-SYSTEM

Gdańsk ul. K. Marksa 169
tel. (058) 41–19–01
proponuje:

- rewelacyjny system TURBO 2000 przyśpieszający współpracę z dowolnym magnetofonem
- sprzęt mikrokomputerowy ATARI XL/XE
- interfejsy magnetofonu, centronics, TOP DRIVE, HAPPY WARP
- sprzęt mikrokomputerowy ATARI ST
- stacja dysków 5,25"
- doskonałe, krajowej produkcji monitory wielosystemowe do ST (niskiej, średniej, wysokiej rozdzielczości) w cenie 299.000,- zł
- sprzęt mikrokomputerowy COMMODORE

ZADZWOŃ, NAPISZ, PRZYJEDŹ I ZOBACZ

(SB-49)

ATASERW

43-100 TYCHY ul. Lencewicza 46/3
tel. 27 69 66

oferuje świetne rozwiązania sprzętowe

do ATARI XL/XE:

1. TURBO DOS—wspaniały DOS na kartridżu
2. TOP DRIVE—do stacji 1050, LDW 2000, CALIFORNIA samodzielny montaż—wysyłkowo (rec. INFORMIK III/88)
3. INTERFEJS CENTRONIKS
4. ROZSZERZENIA PAMIĘCI
5. BASIC XE—kartridż
6. TURBO DOS + BUG65—kartridż

12 miesięcy gwarancji. Informacje i zamówienia telefonicznie (wtorek 8—12, środa, czwartek 16—18) i listownie po otrzymaniu koperty zwrotnej.

K—236

COMPUTER SERVICE

MS elektronik
naprawy komputerów:
Spectrum 48k, +, 128, +2, +3
Amstrad- Schneider
Sharp
Drukarki, Interfejsy
Wyjścia monitorowe
Czynne: od 9.00—16.00
MSelektronik Legionowa 23, 00-343 Warszawa
Dojazd: 105, 305, F (jelonki)
tel 37-76-65.

(K-118)

CZĘŚCI ELEKTRONICZNE

pochodzenia zagranicznego za złotówki układy TTL, LS, CMOS, stabilizatory, mikroprocesory, pamięci.
Krótkie terminy dostaw, atrakcyjne ceny.
Pośredniczymy także przy sprzedaży.
„MARITEX" tel. 22-02-89
Gdynia, ul. Batalionów Chłopskich 3
tlx 054622

K-190

ZX SPECTRUM, TIMEX, ATARI

- **programy użytkowe, edukacyjne, gry**
- **programy dla rzemiosła (receptury, kalkulacje, remanenty)**
- **instrukcje do programów**
- **informacje po nadesłaniu koperty zwrotnej ze znaczkiem**
- **wysyłka na cały kraj — rachunki**

„P.K.T.S." Studio Komputerowe
00-103 Warszawa
ul. Królewska 43 m 25

(SB-59)

Agencyjny Zakład Usługowy SPHW poleca usługi w zakresie:
— serwis komputerów SPECTRUM, C-64, TIMEX, XT, AT, rozszerzanie pamięci
— przestrajanie PAL-SECAM
— wejścia monitorowe TV
Warszawa ul. Puławska 102
tel. 44-87-89
czynny 12.00—19.00
Rachunki, gwarancja
ZAPRASZAMY

D-114

MÓZGOPROCESOR! — rewelacyjną polską grę przygodową dla ZX Spectrum (program + kaseta + opis 6100 zł) otrzymasz pisząc: COMPUTER ADVENTURE STUDIO, Bochnia 32-700, ul. Kazimierza Wielkiego 37/45.

(SB 20)

Wymienię, odstąpię programy, ATARI XL/XE. Informacje gratis. Jakub Przybyłowicz, ul. Murarska 8/14, 27-200 Starachowice.

(SB 62)

STUDIO KOMPUTEROWE

HIT – BIT

programy, literatura, osprzęt komputerowy
ATARI COMMODORE
SPECTRUM AMSTRAD
Warszawa: D.H. „Merkury"
przy Pl. Komuny Paryskiej,
tel. 39-92-51 w. 69
Wysyłka: HIT — BIT
00-967 Warszawa 86
skr. poczt. 168
katalog po załączeniu koperty.

D-51

SERWIS KOMPUTERÓW TEST

40–164 Katowice ul. Modrzewiowa 24/33
poleca naprawy:
ATARI 600, 800, 65, 130, XL, XE
COMODORE 16, 116, +4, 64, 128
DISC DRIVE 1541, 1570, 1571, 1050
MAGNETOFONY COMODORE
DRUKARKI
godz. 9–11, 16–18.

(SB 30)

TOMBAT XL ATARI XE

WYPOŻYCZALNIA TOMBAT XL ATARI XE

— gry i programy użytkowe
— co piąty program gratis
— inne bonifikaty
— opisy gier i instrukcje
— pomoc dla początkujących
— wysyłka na cały kraj
— katalog gratis

Nasz adres:
ul. Magistracka 27 m 26
01-413 Warszawa
Tel. 363-078 godz. 12-20
Zapraszamy!

(SB 12)

---

joy

JOYSTICKI do Atari Commodore, Spectrum, Amstrad precyzyjny mechanizm specjalne styki 9800,– zł, 6 m-cy gwarancja interface do Spectrum

Wysyłka natychmiastowa za zaliczeniem pocztowym

Dla instytucji rachunki płatne przelewem

ELEKTROMECHANIKA

ul. Cegielniania 17
32-410 DOBCZYCE

G-129

UWAGA UŻYTKOWNICY ZX SPECTRUM!

MULTI–DATA

Rewelacyjna Baza Danych!
25 tysięcy znaków
Wysoka korekcja rekordów i umieszczanie nowych pól w dowolnym miejscu przy automatycznej modyfikacji zbioru;
Brak ograniczeń w ilości pól i optymalne rozwiązanie wyszukiwania danych.
Oferujemy również wersje indywidualne opracowane wg. potrzeb użytkownika.
Rozpowszechnianie:
Studio Komputerowe „ZET"
D.H. „SEZAM" II p.
W-wa ul. Marszałkowska 126
UPT 66 skr. p. 14
Wysyłka na cały kraj

D-140

Zakład Produkcyjno-Usługowy Studio Komputerowe „ELDER"

52-126 Wrocław 18, skrytka pocztowa 16
poleca:

1 — **MIKROEXPRESS'88 — kwartalnik komputerowy na kasecie magnetofonowej C-60. W każdym numerze ok. 500KB informacji i programów dla użytkowników ZX SPECTRUM.**
Do nabycia nr 1 i 2.
Cena 6000 zł (bez kosztów kasety i opłaty pocztowej)
2 — **Nagrane na taśmę magnetofonową (kasetę) instrukcje w języku polskim do programów edukacyjnych, użytkowych i gier na ZX SPECTRUM.**
Cena 500 zł za 1 instrukcję (bez kosztów kasety i opłaty pocztowej).

(SB 63)

Rewelacyjnie niskie ceny!!! Programy dla Atari XL/XE. Informacje (koperta + znaczek) Alek i Robert Corp, ul. Zielona 14 lub ul. Niecała 9, 36-100 Kolbuszowa.

(Sd 303)

SAM WYKONASZ OBWODY DRUKOWANE
Zestaw (laminat, odczynniki, instrukcja)
Cena 1430 zł. Wysyłka za zaliczeniem pocztowym
Zamówienia kierować: A. Kaw[illegible]
Łódź 1, skrytka pocztowa 344
Płatne przy odbiorze
ZAWSZE AKTUALNE

D-148

ATARI TURBO 2000 F

Nowy system transmisji danych z magnetofonem przyspieszony do 6700 bodów

Komplet:
— cartridge
— kaseta z 5 programami kopiującymi
— przeróbka magnetofonu
— 12 m-cy gwarancji za 29 000 zł

oraz interfejs do standardowego magnetofonu za 22 000 zł oferuje firma MUEL ul. Cząstkowska 30, 01-678 Warszawa tel. 33-40-91

D-121

Programy , pamięci

VC-20

B.U.K. "MAVIC"
ul. Karasia 18
64-100 Leszno

Drogi Bajtku!

Na listy czytelników odpowiada DOMINIK FALKOWSKI

Od niedawna jestem posiadaczem Spectrum+. Przeglądając różne materiały na jego temat, trafiłem na kilka informacji, których nie potrafię do siebie dopasować. Po pierwsze, to czy został naciśnięty jakiś klawisz jest sprawdzane przez program zapisany w ROM-ie. Po drugie, ROM Spectrum+ i zwykłego Spectrum podobno niczym się nie różni. Jak więc sprawdzane jest, czy nie został naciśnięty któryś z nowych klawiszy?

M. Bujak, Rzeszów

Wszystkie znalezione przez Pana informacje są prawdziwe. Rozwiązanie tego paradoksu polega na tym, że zmieniony został układ elektryczny klawiatury. W zwykłym Spectrum pod klawiaturą znajdowała się dwuwarstwowa folia, i naciśnięcie dowolnego klawisza powodowało zwarcie dwóch nadrukowanych na tą folię przewodzących ścieżek. W nowym Spectrum zastosowano rozwiązanie nieco podobne, ale folia jest już trójwarstwowa, toteż naciskanie nowych klawiszy powoduje zwarcie nie jednej pary ścieżek, ale dwóch par. W efekcie procedura zawarta w ROM-ie podczas odczytywania stanu klawiatury „zauważa" dwa naciśnięte klawisze, i rozpoznaje odpowiedni znak tak, jakby użytkownik rzeczywiście użył dwóch klawiszy. Proszę sprawdzić, że mimo iż pewne znaki zostały na klawiaturze pominięte, można je wprowadzać tak jak w zwykłym Spectrum, np. cudzysłów jako shift-P.

Czy można samemu zrobić klawiaturę do Spectrum? Od trzech miesięcy czekam na wymianę folii spod klawiatury w zakładzie naprawczym, i ciągle słyszę odpowiedź — nie ma dostaw. Podobno na szynie krawędziowej są dostępne wszystkie potrzebne wyprowadzenia, ale jak z nich skorzystać?

R. Stojek, Warszawa

Poruszony przez Pana problem jest w pewnym sensie tak stary jak samo Spectrum — od początku wiadomo było, że najsłabszą częścią całego komputera jest klawiatura — tania, ale niezbyt trwała. Dwa, trzy lata temu kilka brytyjskich firm robiło dobre interesy, sprzedając (za cenę często niewiele odbiegającą od ceny nowego komputera) profesjonalne klawiatury do Spectrum. A w naszych warunkach — trzeba sobie radzić samemu. Nie będziemy na naszych łamach drukować opisu takiej konstrukcji, ale radzę poszukać Radioelektronika nr 7/1985 — o ile wiem, tam znajdzie Pan potrzebną pomoc.

Czy istnieje szybsza procedura dzielenia liczby z przedziału 0-65535 przez 10 dla procesora Z80? Podobno istnieją specjalizowane procedury do dzielenia przez określoną liczbę. Jeśli to możliwe proszę o jej treść.

```
;Wejście:     HL      — dzielna
;             C       — dzielnik
;Wyjście:     HL      — iloraz;
;             A       — reszta;
100 DIV21     LD      B,16
110           XOR     A
120           ADD     HL,HL
130 LOOP      RLA
140           SUB     C
150           JR      NC,JUMP
160           ADD     A,C
170 JUMP      CCF
180           ADC     HL,HL
190           DJNZ    LOOP
200           RET
```

Jerzy Ziaja, Sieraków Śląski

Rzeczywiście, nie znam szybszej procedury niż napisana przez Pana, więc drukujemy ją jako propozycję do wykorzystania w swoich programach dla tych z naszych Czytelników, którzy korzystają z asemblera na Z80. (Uwaga — treść procedury podajemy na odpowiedzialność autora, nie była przez nas szczegółowo testowana.) A może ktoś zna inny, szybszy algorytm rozwiązujący to samo zadanie? Jeżeli tak, to przyślijcie nam do redakcji, chętnie go opublikujemy.

Jestem posiadaczem mikrokomputera ZX Spectrum Plus. Interesuje mnie grafika. Proszę o wyjaśnienie pewnej kwestii dotyczącej atrybutów. Otóż nie wiem, jak umieścić w jednym polu (8x8 pixeli) kilka kolorów jednocześnie. Jest to realizowane w większości plansz tytułowych gier.

Paweł Burakowski, Katowice

Kolory obrazu Spectrum przechowywane są w 768 bajtach pamięci, od adresu 22528 do 23295. Każdemu polu 64 pixeli przypisana jest liczba ośmiobitowa, atrybut pola. Zapalenie lub zgaszenie odpowiednich bitów odpowiada za stan migotania, podwyższonej jaskrawości pola, oraz kolor liter i tła. Tak więc w obrębie jednego pola można uzyskać tylko dwa kolory: liter (ink) i tła (paper).

Z poziomu Basica realizacja jest prosta.

PRINT INK 6; PAPER 1; „Bajtek"

spowoduje wyświetlenie napisu „Bajtek" na niebieskim tle, kolorem żółtym. Teraz

PLOT INK 0;1,175

sprawi, że w lewym górnym rogu pojawi się czarny punkt. Jego atrybut zostanie jednak narzucony wszystkim innym pixelom w obrębie pola 8x8 i napis „Bajtek" zmieni kolor.

Screeny, czyli obrazki do gier wymagają ogromnego nakładu pracy. Nikt nie rysuje ich z poziomu Basica; używa się zwykle programów typu Art Studio lub The Artist II. często stosowana sztuczka z mieszaniem kolorów polega na tym, że dzieli się pole 8x8 na dwie części wymaganą linią. Jedną z tych części zaczernia się, zaś druga pozostaje biała. Po nałożeniu atrybutów INK-PAPER pole to wygląda, jakby składało się z dwóch części o różnych kolorach.

W ten sposób wykonana jest np. tęcza, znak firmowy Spectrum. Przypatrz się uważnie screenowi podczas wczytywania. Zanim zostanie on pokolorowany, treść jego jest raczej nieczytelna, widać „wyłażące" kwadraty. Dopiero po pokryciu treści atrybutami, obrazek jest czytelny.

Jestem posiadaczem mikrokomputera Timex 2048. Piszę właśnie program, który chciałbym zabezpieczyć przed piratami. Nie wiem, jak to zrobić. (...)

Adam Foland, Wesoła

Sytuacja nie jest prosta. Żadne bowiem zabezpieczenie nie przeszkodzi piratom tak skutecznie, jak zrobiłoby to prawo autorskie programu. Pisaliśmy nieraz w „Bajtku", jak można próbować zabezpieczać programy, lecz nawet najwymyślniejsza blokada nie oprze się zręcznemu hackerowi.

Obecnie na Zachodzie rezygnuje się z zabezpieczeń, gdyż pochłania to masę czasu i pieniędzy, sprawiając w dodatku kłopot użytkownikowi. Rezygnacja z zabezpieczeń pozwala na bardzo szybkie rozprzestrzenianie się programu drogą nielegalną, lecz sam program jest mało wart. Nie zakupiony u producenta pozbawiony jest instrukcji, która jest do programu niezbędna. Posiadacz bezprawnej kopii nie ma też łatwego dostępu do nowych wersji programu, wciąż ulepszanego przez producenta.

Mowa tu o programach, które kosztują setki dolarów i są towarem dużo ważniejszym od hardware-u. Jeśli chodzi o gry, to prawie wszystkie zapisywane są w systemie „Speed-Lock", na którym „wysiadają" nasze kopiery. To jednak nie przeszkoda. Działa bowiem tabun pseudo-hackerów, którzy wciśnięciem przycisku w przystawce nagrywają grę na taśmę, dorabiają napis „Broken by ... i sprzedają grę. Często niszczy się w ten sposób program obsługi muzyki, początkowy obrazek lub nawet jedno z menu!

Jak więc nie byłby zabezpieczony Twój program, zawsze znajdzie się ktoś, kto odblokuje go szybciej, lub wolniej.

Po przeczytaniu artykułu „Modem — co to jest?" zainteresowaliśmy się bliżej tym urządzeniem. Chcielibyśmy dowiedzieć się, jak kształtują się ceny modemów do Atari i Spectrum, gdzie można je nabyć i jak podłączyć je do ZX Spectrum. Czy wszystkie programy z Atari będą prawidłowo funkcjonowały na Spectrum i odwrotnie?

Romuald Stefański, Marcin Kowalczyk, Tychy

Modem jest to urządzenie służące do transmisji danych między komputerami za pomocą linii telefonicznej. Sposób transmisji jest ustalony i za pomocą modemów każdy komputer może komunikować się z każdym.

Chodzi jednak tylko o transmisję danych. W szczególności danymi może być kod programu, gry itp. Najczęściej jest to plik tekstowy do edytora, dane do arkusza obliczeniowego lub pojedyńcze ciągi znaków — pytania i odpowiedzi. Nie ma mowy o tym, by np. przesłać program ze Spectrum na Atari i uruchomić go! Tu chodzi o komunikację, a nie ukompatybilnienie.

O ile nam wiadomo, nie produkuje się modemów do Spectrum. Prawdopodobnie istnieją modemy pozwalające na komunikację za pomocą linii telefonicznej, lecz w systemie CP/M, który dostępny jest na Spectrum dzięki stacji dysków Timex FDD 3000. Po drobnych przeróbkach można wymieniać oprogramowanie z innymi komputerami pracującymi pod CP/M. Stacja posiada dwa kanały transmisji RS 232 i prawdopodobnie one wykorzystywane są przez modem. Bliższych informacji nie posiadamy.

Modem do Atari kosztuje ok. 200.000 zł i rozprowadzany jest przez firmę Karen. Homologacja i opłaty pocztowe mogą być dokonane bez trudności, gdyż do każdego modemu dołączane są odpowiednie dokumenty. Wiele informacji o modemie do Atari możecie znaleźć oczywiście w Klanie Atari.

Zwracam się z prośbą o pomoc, radę, gdzie mógłbym nabyć programy dla mikrokomputera Meritum. Najbardziej interesują mnie gry, chciałbym kupić taśmy. (do listu dołączona była lista tytułów gier — red.)

Cezary Kramarczyk, Międzyrzecz

Meritum jest produktem typowo polskim. Dlatego nie ma nań prawie żadnego oprogramowania, nie mówiąc już o profesjonalnym, którego nie ma w ogóle. Niektóre tytuły z załączonej przez Pana listy wydają mi się znajome, być może są to adaptacje prościutkich gier ze Spectrum. Pozostałe napisane są prawdopodobnie przez rodzimych autorów. Nie jest nam wiadome, kto zajmuje się rozpowszechnianiem programów na Meritum. Proszę zwrócić się do Klubu Meritum, jednego z prezentowanych w „Bajtku".

Zostałem obdarowany komputerem ZX Spectrum 128 oraz drukarką DMP 2000. Komputer posiada gniazdo interfejsu RS-232 do drukarki. Gdzie mógłbym szukać pomocy technicznej aby połączyć drukarkę z komputerem tak, aby możliwe było korzystanie z wersji języka Basic 128? Zakłady usługowe montują i sprzedają wprawdzie interfejsy, lecz wówczas zapewniają działanie drukarki jedynie w trybie Basic 48.

Sławomir Butkiewicz
ul. 1-go Maja 17/59
16-400 Suwałki

Nie ma wśród nas speca od sprzętu, który mógłby Ci coś poradzić. Spectrum 128 jest jednak już nieźle rozpowszechniony w Polsce i zapewne niejeden z majsterkowiczów stawał przed tym problemem, co nasz Czytelnik. Jeśli tak, to prosimy o pomoc w jego imieniu. Oczekujemy też na wyniki podłączenia, w celu publikacji w „Bajtku". Nie jest to bowiem jedyny list, jaki dostaliśmy w ciągu ostatniego miesiąca w sprawie Spectrum 128 i drukarki. Czy Basic 128 oferuje aż tak wielkie możliwości, że warto poczynić przeróbkę w celu podłączenia drukarki?

TYLKO DLA PRZEDSZKOLAKÓW

TEN OKROPNY BASIC

Programowanie strukturalne to — mówiąc obrazowo — konstruowanie klocków, z których następnie budujemy cały program. Każdy, kto próbował pisać programy w LOGO przekonał się, jak wygodna a przede wszystkim przejrzysta jest ta metoda.

Z klocków najlepiej buduje się domy. Wie o tym każdy przedszkolak. My też zabawimy się w budowniczych. Oczywiście — jak przystało na nowoczesne budownictwo — nasz dom będzie składał się z pewnych typowych elementów budowlanych. Dla każdego z nich napiszemy osobny podprogram. Aby więc uniknąć pomyłek musimy cały program dokładnie opisać.

W pierwszej kolejności zaprojektujemy elementy do budowy ścian.

```
1000 REM *** okno duże ***
1010 PRINT "!oo!";
1020 RETURN
1100 REM *** okno małe ***
1110 PRINT "! o!";
1120 RETURN
1200 REM **** balkon *****
1210 PRINT "!==!";
1220 RETURN
1300 REM ***** drzwi *****
1310 PRINT "! #!";
1320 RETURN
```

Następnie fragmenty dachu.

```
1400 REM ***** dach ******
1410 PRINT "____";
1420 RETURN
1500 REM * dach i antena *
1510 PRINT "_\/_";
1520 RETURN
```

Z takich klocków możemy poskładać przeróżne domy. Rozpocznijmy od zaprojektowania poszczególnych kondygnacji.

```
500 REM ***** parter *****
510 GOSUB 1000: REM okno duże
520 GOSUB 1200: REM balkon
530 GOSUB 1300: REM drzwi
540 GOSUB 1000: REM okno duże
550 PRINT
560 RETURN
600 REM ***** piętro *****
610 GOSUB 1000: REM okno duże
620 GOSUB 1200: REM balkon
630 GOSUB 1100: REM okno małe
640 GOSUB 1000: REM okno duże
650 PRINT
660 RETURN
700 REM ****** dach ******
710 GOSUB 1400: REM dach
720 GOSUB 1500: REM dach i antena
730 GOSUB 1400: REM dach
740 GOSUB 1400: REM dach
750 PRINT
760 RETURN
```

Mamy już więc wszystko co jest nam potrzebne. Teraz już możemy układać poszczególne kondygnacje. Rozpocznijmy trochę inaczej niż na prawdziwej budowie — od dachu.

```
10 GOSUB 700: REM dach
20 GOSUB 600: REM piętro
30 GOSUB 600: REM piętro
40 GOSUB 600: REM piętro
50 GOSUB 600: REM piętro
60 GOSUB 500: REM parter
100 GOTO 100: REM koniec programu
RUN
      \/
TooTT==TT oTTooT
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! #!!oo!
```

Opisy w programie przydatne są nie tylko podczas pisania programu, ale także wtedy, gdy chcemy go przerobić. Nasz program został napisany w taki sposób, że bardzo łatwo można go wykorzystać do rysowania innych domów. Jeśli na przykład chcemy mieć wyższy dom wystarczy dopisać kilka linii.

```
21 GOSUB 600: REM piętro
22 GOSUB 600: REM piętro
23 GOSUB 600: REM piętro
24 GOSUB 600: REM piętro
RUN
      \/
TooTT==TT oTTooT
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! o!!oo!
!oo!!==!! #!!oo!
```

Możemy także narysować zupełnie inny dom. Wymaga to jednak większych zmian. Zmodyfikujemy nieco podprogramy opisujące kondygnacje.

```
500 REM ***** parter *****
510 GOSUB 1200: REM balkon
520 GOSUB 1300: REM drzwi
530 GOSUB 1000: REM okno duże
540 GOSUB 1000: REM okno duże
550 GOSUB 1300: REM drzwi
560 GOSUB 1200: REM balkon
570 PRINT
580 RETURN
600 REM ***** parter *****
610 GOSUB 1200: REM balkon
620 GOSUB 1300: REM drzwi
630 GOSUB 1000: REM okno duże
640 GOSUB 1000: REM okno duże
650 GOSUB 1300: REM drzwi
660 GOSUB 1200: REM balkon
670 PRINT
680 RETURN
700 REM ****** dach ******
710 GOSUB 1400: REM dach
720 GOSUB 1500: REM dach i antena
730 GOSUB 1400: REM dach
740 GOSUB 1400: REM dach
750 GOSUB 1500: REM dach i antena
760 GOSUB 1400: REM dach
770 PRINT
780 RETURN
```

Ustawienie kondygnacji możemy pozostawić takie, jak w poprzednim programie.

```
RUN
      \/                  \/
T==TT oTTooTTooTT oTT==T
!==!! o!!oo!!oo!! o!!==!
!==!! o!!oo!!oo!! o!!==!
!==!! o!!oo!!oo!! o!!==!
!==!! o!!oo!!oo!! o!!==!
!==!! o!!oo!!oo!! o!!==!
!==!! o!!oo!!oo!! o!!==!
!==!! o!!oo!!oo!! o!!==!
!==!! #!!oo!!oo!! #!!==!
```

Okazuje się więc, że programowanie strukturalne w BASIC-u jest możliwe. Nie jest oczywiście tak łatwe i naturalne jak w LOGO, niemniej jednak przy odrobinie wyobraźni da się zrealizować. Nie warto więc powtarzać pochopnych i nieprzemyślanych sądów. I dotyczy to nie tylko zagadnień z dziedziny informatyki.

Na co zwraca Wam uwagę

Romek

GALERIA BAJTKA

dokończenie ze str.32



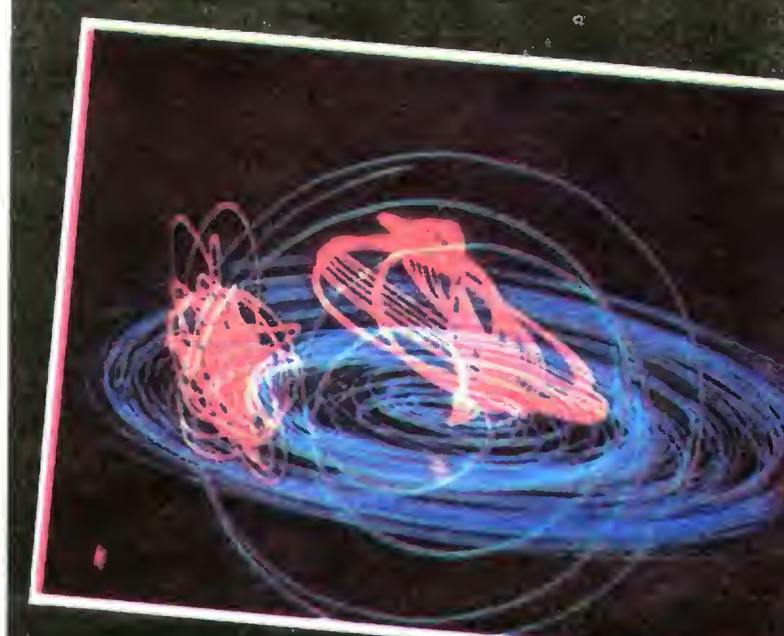
I od tej pory, na niezliczonych wystawach i pokazach, ujawnia potęgę światła. Przyniosło mu to członkostwo Royal Society of Arts w Londynie i innych świetnych towarzystw. Został laureatem wielu prestiżowych nagród.

Jego pokazy dostarczają widzom niezapomnianych przeżyć. Synchronizacja obrazów laserowych z muzyką porusza najbardziej czułe struny w człowieku, sięgające do nieuświadamianych sobie na codzień naszych związków z kosmosem. Podjął Ostoja-Kotkowski wysiłek „przetłumaczenia na język laserowy" muzyki Krzysztofa Pendereckiego. W ramach przygotowywania „wyobrażeń laserowych" interesują go w jednakowym stopniu dźwięki z przestrzeni kosmicznej, jak też odgłosy z dna morza, a szczególnie sygnały, którymi porozumiewają się delfiny.

W 1987 roku przeprowadził eksperyment SOLARIS, będący „kinetycznym freskiem słonecznym" czyli zdarzeniem artystycznym opartym na wykorzystaniu słońca jako źródła światła regulowanego, za pomocą którego tworzy się kolory i kształty. Słońce steruje również w tym eksperymencie syntezatorem dźwięku. SOLARIS był dalszym krokiem w wysiłkach artysty dążącego do połączenia obrazu i dźwięku.

Już w 1968 roku, korzystając z komputera CONTROL DATA 6400, tworzy Stanisław Ostoja-Kotkowski pierwszy wiersz komputerowy, zatytułowany „Przypływ żywiołu". Pierwszą swoją grafikę komputerową wykonał w Bell Research Laboratories (New Jersey, USA) w 1967 roku. Dopiero jednak zaproponowana przez prof. Benoit B. Mandelbreta (ur. w 1924 roku w Warszawie) metoda „fraktali" uznana została przez Stanisława Ostoję-Kotkowskiego za rzeczywiście godną uwagi. Prezentujemy obok ostatnią pracę Ostoi—Kotkowskiego opartą na systemie Mandelbreta. Nazwał ją „ID" (według „Oxford dictionary": instynktowny impuls indywidualny), gdyż przy tworzeniu tej grafiki kierował się wyłącznie instynktem. Uznał jednak wynik za tak ciekawy, że pozostawił go w pamięci komputera.

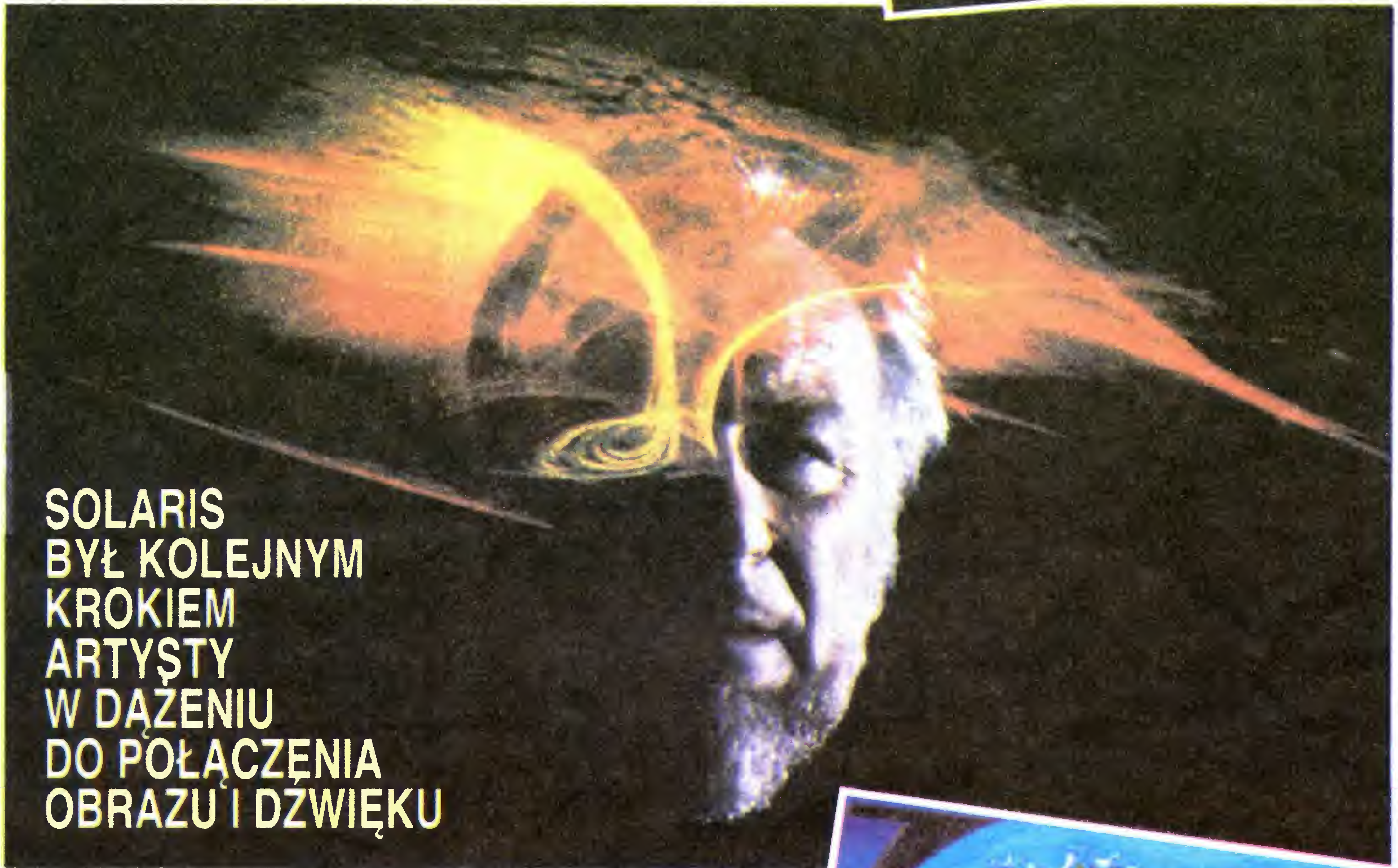
W. Siwiński



POTĘGA ŚWIATŁA

„Wyobrażenia laserowe"





SOLARIS BYŁ KOLEJNYM KROKIEM ARTYSTY W DĄŻENIU DO POŁĄCZENIA OBRAZU I DŹWIĘKU

Stanisław Ostoja-Kotkowski



„ID" czyli instynktowny impuls indywidualny

Po opuszczeniu położonego wśród zielonych wzgórz pod Adelajdą domu Stanisława Ostoi-Kotkowskiego pozostaje odczucie obcowania z magikiem. Pod powiekami dalej trwają efekty jego wyobrażeń wizualnych, w uszach dalej słychać nieziemskie dźwięki...

Stanisław Ostoja-Kotkowski jest jednym z najwybitniejszych artystów Australii, ale trudno byłoby jednoznacznie stwierdzić kim jest: malarzem, grafikiem czy rzeźbiarzem. Fascynuje go przestrzeń, światło i muzyka — a mówiąc jeszcze precyzyjniej: możliwości połączenia tych wszystkich elementów. Eksperymentuje w tym zakresie od lat, korzystając z najnowocześniejszych narzędzi. „Korzystając" to zresztą słabo powiedziane, bo Ostoja-Kotkowski rzuca się łapczywie na wszystko, co dostarcza najnowsza technologia. Jako pierwszy na świecie użył w 1968 roku lasera zamiast pędzla. Gdy tylko komputery przekroczyły próg szerokiej dostępności — skorzystał natychmiast z komputerów.

Urodził się w 1922 roku w Gołubiu. Przed wojną zdążył jeszcze zrobić małą maturę w Przasnyszu. W 1945 Niemcy zabrali go z sobą jako zakładnika. Wyzwolony został przez wojska amerykańskie, a następnie ukończył Akademię Malarską w Dusseldorfie. W roku 1950 zaczął myśleć o wyjeździe w daleki świat. Zdecydował się na Australię, ponieważ jej rząd opłacał podróż i oferował pomoc w urządzeniu się na miejscu. Przebijał się przez to nowe życie, jak wszyscy emigranci, mniejsza o szczegóły, ważne że znalazł swoje miejsce.

Australia — kraj, w którym żyje i tworzy Ostoja-Kotkowski — pobudza jego wyobraźnię. Australia to bowiem dla artysty tysiące kilometrów kwadratowych niezwykłych kolorów, światła, atmosfery. „To właśnie we wnętrzu tego kraju — mówi Ostoja-Kotkowski — odkryłem potęgę światła."

Zafascynowała go eksplozja barw, której źródłem były palące promienie australijskiego słońca. Eksperymentował z fotografią, z malowaniem przy użyciu farb polimerowych... Za swe osiągnięcia uzyskał stypendium im. Churchila. Pozwoliło mu to na odbycie wielomiesięcznej podróży, w trakcie której odwiedził Europę, m.in. Polskę, Japonię, wreszcie USA. Tam właśnie, na Uniwersytecie Kalifornijskim, zademonstrowano mu doświadczenia z laserem. „Doznałem wówczas olśnienia — mówi. — To moja wielka nadzieja, pomyślałem. To nareszcie możliwość wyzwolenia wszystkich najwspanialszych kolorów natury."

dokończenie na str.31